# University of Southern California
# Viterbi School of Engineering


# EE577A Spring 2019

# VLSI System Design

# General-Purpose Microprocessor Design & BNN Accelerator

**Instructor: Prof. Pierluigi Nuzzo**

*Shang Zhou 2725583204*

*Anh Vu 2396504143*

*Hongxiang Gao 8095639536*

May 3, 2019

# Contents


**1 General-Purpose CPU Design** **2**
- 1.1 Instruction Decoding Stage . . . 2
  - 1.1.1 Schematic Design . . . 2
  - 1.1.2 Layout Design . . . 3
  - 1.1.3 Block Simulation . . . 4
  - 1.1.4 LVS Result . . . 5
- 1.2 Execution Stage . . . 19
  - 1.2.1 5 bit Multiplier . . . 19
  - 1.2.2 Shifter . . . 24
  - 1.2.3 Adder . . . 30
  - 1.2.4 And Logic . . . 36
  - 1.2.5 Or Logic . . . 42
- 1.3 Memory Stage . . . 48
  - 1.3.1 Schematic Design . . . 48
  - 1.3.2 Layout Design . . . 48
  - 1.3.3 LVS Result . . . 49
- 1.4 Stage Registers . . . 52
  - 1.4.1 Schematic Design . . . 52
  - 1.4.2 Layout Design . . . 53
  - 1.4.3 LVS Result . . . 53
- 1.5 CPU . . . 65
  - 1.5.1 Schematic Design . . . 65
  - 1.5.2 Layout Design . . . 65
- 1.6 Python Code List . . . 66
  - 1.6.1 Front-End Part . . . 66
  - 1.6.2 Back-End Part . . . 88

**2 Bayesian Neural Network (BNN)** **90**
- 2.1 Schematic Design . . . 90
- 2.2 Layout Design . . . 91
- 2.3 LVS Result . . . 91
- 2.4 Python Code List . . . 93

**3 Simulation Result** **96**
3.1 CPU Functionality Simulation . . . . . 96
3.1.1 Input Instructions List . . . . . 96
3.1.2 Virtuoso Simulation Result . . . . . 96
3.1.3 Back-End Result . . . . . 104
3.2 BNN Functionality Simulation . . . . . 105
3.2.1 Output of BNN Based GRNG . . . . . 105
3.2.2 Simulation Result . . . . . 105
3.2.3 Back-End Result . . . . . 105
3.3 CPU & BNN Performance Comparison . . . . . 107
3.3.1 Simulation result . . . . . 113
3.4 CPU Parameter Conclusion . . . . . 114

**4 Conclusion** **115**

**References** **116**

# List of Figures


1.1 ID stage schematic . . . . . 2
1.2 ID stage layout . . . . . 3
1.3 Simulation result for front-end block overview . . . . . 4
1.4 Simulation result for front-end block . . . . . 5
1.5 5bit Multiplier schematic . . . . . 19
1.6 5bit Multiplier Layout . . . . . 20
1.7 5bit Multiplier Simulation . . . . . 20
1.8 Shifter schematic . . . . . 24
1.9 Shifter Layout . . . . . 25
1.10 Shift Left Simulation . . . . . 26
1.11 Shift Right Simulation . . . . . 26
1.12 Adder schematic . . . . . 30
1.13 Concept of Change an Adder to a Subtractor[1] . . . . . 30
1.14 Adder Layout . . . . . 31
1.15 Adder Simulation . . . . . 32
1.16 Min Block Simulation . . . . . 32
1.17 And schematic . . . . . 36
1.18 And Layout . . . . . 37
1.19 And Simulation . . . . . 38
1.20 And Logic Cell Optimization . . . . . 41
1.21 And logic Optimization . . . . . 42
1.22 OR schematic . . . . . 42
1.23 Or Layout . . . . . 43
1.24 Or Simulation . . . . . 44
1.25 Or Cell Optimization . . . . . 47
1.26 Or logic Optimization . . . . . 47
1.27 Memory schematic . . . . . 48
1.28 Memory layout . . . . . 48
1.29 Stage Register of ID+EX, EX+MEM, and MEM+WB schematic. . . . . 52
1.30 Id +Ex Stage Layout . . . . . 53
1.31 Ex+Mem Stage Layout . . . . . 53
1.32 Mem+WB stage layout . . . . . 53
1.33 CPU Schematic . . . . . 65
1.34 CPU layout . . . . . 66

2.1 BNN schematic . . . . . 90
2.2 BNN layout . . . . . 91

3.1 Virtuoso Simulation Result of CPU with ISA Input . . . . . 104
3.2 Virtuoso Simulation Result of BNN Schematic . . . . . 106
3.3 Virtuoso Simulation Result of BNN Layout . . . . . 106
3.4 Virtuoso Simulation Result of CPU with BNN Input . . . . . 113

# List of Tables

3.1 CPU Parameter Conclusion . . . . . 114

# Introduction

In Part 1 of this project, we implemented a simple CPU with the knowledge we gained in EE577A, including datapath and SRAM designs as well as scripting (Python or Perl) experience from the EE577A labs, and aim at optimizing for area, delay, and power. The Front-end python code will convert the input ISA instructions into a vector file for simulation. The Back-end python code will verify the result which has been generated from our CPU.

In Part 2 of this project, We implemented Bayesian Neural Networks (BNNs) architecture. By introducing weights associated with a conditional probability distribution, BNNs are capable of resolving the overfitting issues commonly seen in machine learning from conventional neural networks. We are going to design hardware modules to speed up variation inference algorithms in BNNs by designing high-performance pipelining in data-path blocks. The BNN accelerator both at the schematic and layout levels. The BNN has five inputs.The data-width of the inputs and weights are 5 bits. Software prototype of the random-number generator has been applied to generate random weights for the BNN. We also use pipelined architecture to support high throughputs. Then, map the BNN operations (weighted input additions) to the CPU from Project Part 1 by developing the assembly code that will be executed by the CPU. The Front-end python code will convert the input ISA instructions into a vector file for simulation.

This report contains 4 sections of EE577a final project. Section 1 shows the design of the efficient general-purpose CPU that supports fundamental instructions( Add, Bitwise operations, Store Word, and Load Word...). In order to achieve high performance, we applied pipelining circuit. Pipelining, which is a standard feature in RISC processors, is much like an assembly line. Because the processor works on different steps of the instruction at the same time, more instructions can be executed in a shorter period of time.[2] Section 2 shows the design a BNN ( Binary Neural Network) Accelerator by using the software and hardware prototype. Section 3 compares the performance of BNN with the general-purpose CPU design in Part1 at the same scenario. Section 4 is the conclusion of our final project and our acknowledgement to the professor and TAs.

# 1. General-Purpose CPU Design

## 1.1 Instruction Decoding Stage

For the Instruction Decoding (ID) stage, we designed register file (RF) by using the 1 to 8 decoder, TG based DFF, MUX/DEMUX.

For the ID stage schematic design, we are using clock-gating to reduce the dynamic power. The following part will show the schematic and layout of our design.

### 1.1.1 Schematic Design


**Figure 1.1:** *ID stage schematic*

### 1.1.2 Layout Design

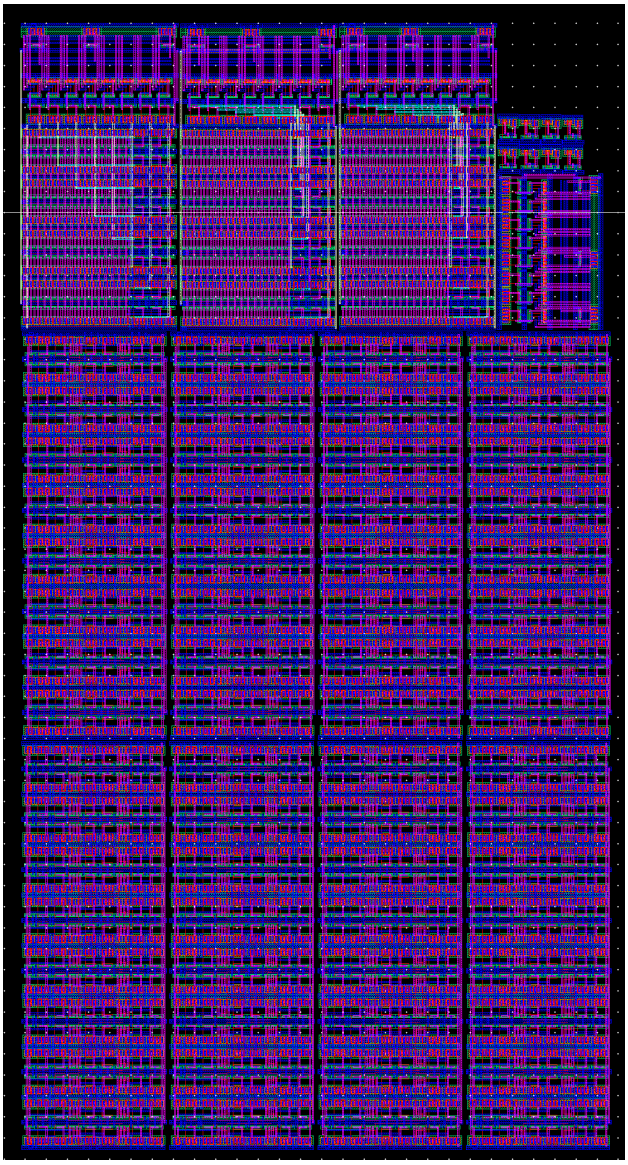

**Figure 1.2:** *ID stage layout*

### 1.1.3 Block Simulation

The section below shows the simulation result for the front-end IF+ID block. We are testing the information passing through the destination register chosen. Figure 1.3 show the overview and 1.3 show a closer look at the data value. The 1st signal is the read select of which register of which MUX we wish to read from, the 2nd signal show the input data of Register1 in MUX1( the test was setup to read from this Register, for ease of demonstration, other registers input were not shown), the 3rd signal shows the output data, and the 4th signal is the clock signal.

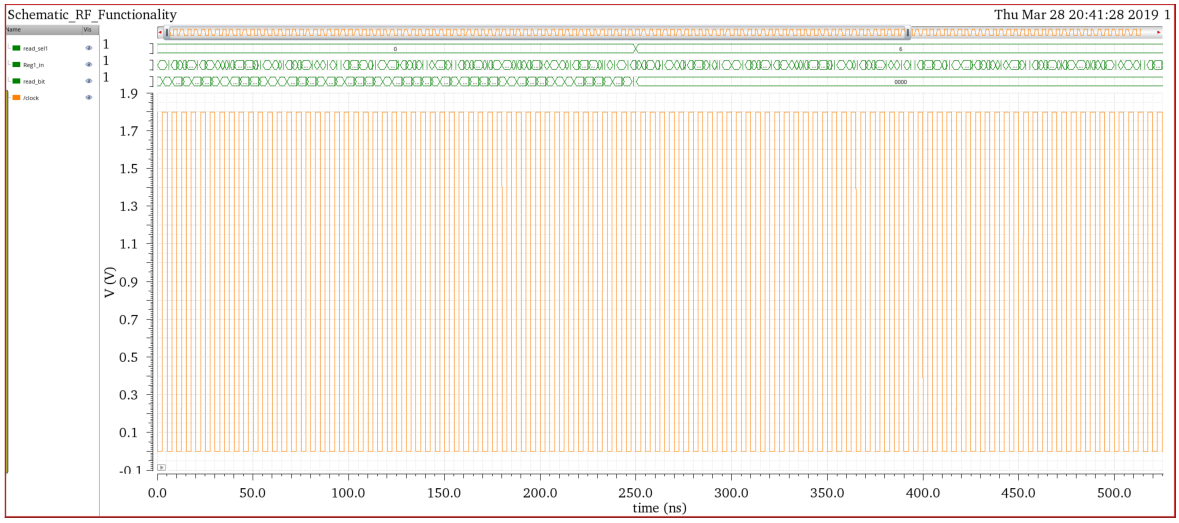


**Figure 1.3:** *Simulation result for front-end block overview*

From 0ns to 250ns, the read select signal is 000 so we chose to read the information from the upper MUX1, and into Register 1. After 250ns, the read select signal is 0h6 in hex or 0b110 in binary, meaning we want to read MUX1 and into Register 7. The output data from 0 to 250ns should duplicate the input data, and after 250ns will be 0 since all the input to Register 7 of MUX1 was set to be 0.

Figure 1.4 show a closer result from 0 to 250ns of how the output data duplicate the input signal from Register 1 of MUX1 after the clock edge.

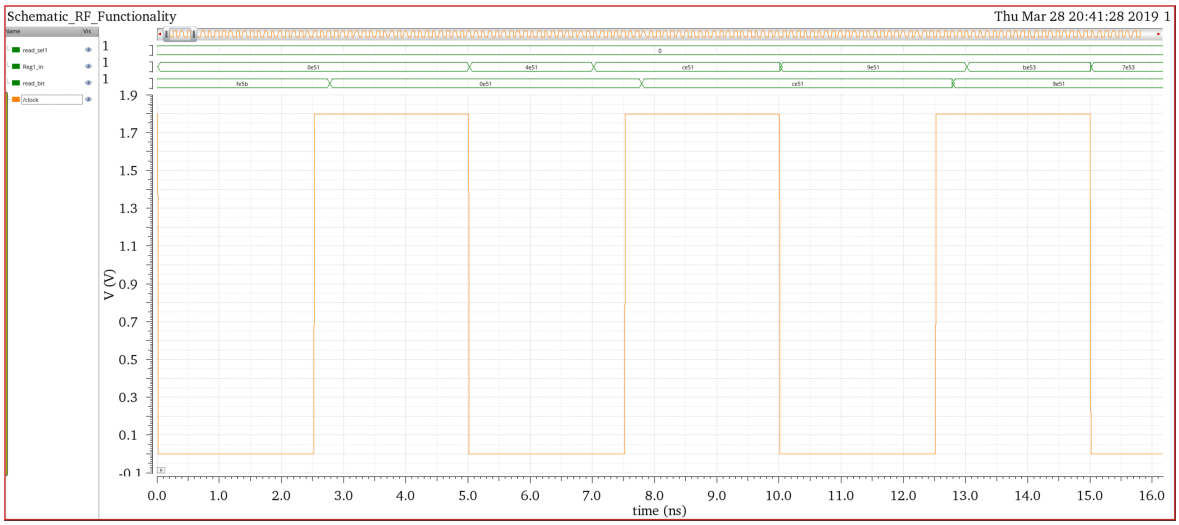


**Figure 1.4:** *Simulation result for front-end block*

### 1.1.4 LVS Result

The section below show the LVS Result for the Front End blocks, which include the register files, 1 to 8 MUX, 8 to 1 DEMUX, and 3 to 8 decoder.

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
         163              nets
         35               terminals
         480              pmos
         480              nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
         163              nets
         35               terminals
         176              pmos
         176              nmos


    Terminal correspondence points
    N149        N50         bit<0>
    N160        N13         bit<10>
    N157        N14         bit<11>
    N154        N15         bit<12>
```

```
    N150        N16         bit<13>
    N144        N17         bit<14>
    N138        N18         bit<15>
    N143        N49         bit<1>
    N137        N5          bit<2>
    N134        N6          bit<3>
    N130        N7          bit<4>
    N162        N8          bit<5>
    N159        N9          bit<6>
    N156        N10         bit<7>
    N153        N11         bit<8>
    N147        N12         bit<9>
    N140        N21         clock
    N131        N1          gnd!
    N148        N4          in<0>
    N151        N29         in<10>
    N145        N30         in<11>
    N139        N31         in<12>
    N135        N32         in<13>
    N132        N33         in<14>
    N128        N19         in<15>
    N141        N3          in<1>
    N136        N2          in<2>
    N133        N22         in<3>
    N129        N23         in<4>
    N161        N24         in<5>
    N158        N25         in<6>
    N155        N26         in<7>
    N152        N27         in<8>
    N146        N28         in<9>
    N142        N0          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                                layout  schematic
                                   instances
        un-matched                 0       0
        rewired                    0       0
        size errors                0       0
        pruned                     0       0
        active                     960     352
        total                      960     352

                                     nets
        un-matched                 0       0
        merged                     0       0
        pruned                     0       0
        active                     163     163
        total                      163     163

                                   terminals
```

```
        un-matched              0       0
        matched but
        different type          0       0
        total                   35      35


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

Code Listing 1.1: 16 bit Register LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
        208             nets
        149             terminals
        276             pmos
```

```
      276              nmos

Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
   count
      208              nets
      149              terminals
      187              pmos
      187              nmos


Terminal correspondence points
N74         N1          gnd!
N174        N98         out_bit<0>
N67         N24         out_bit<10>
N197        N152        out_bit<11>
N181        N151        out_bit<12>
N162        N150        out_bit<13>
N146        N149        out_bit<14>
N129        N148        out_bit<15>
N156        N97         out_bit<1>
N140        N96         out_bit<2>
N124        N106        out_bit<3>
N107        N105        out_bit<4>
N92         N104        out_bit<5>
N73         N103        out_bit<6>
N205        N145        out_bit<7>
N187        N144        out_bit<8>
N169        N138        out_bit<9>
N83         N101        read_sel<0>
N64         N100        read_sel<1>
N195        N99         read_sel<2>
N117        N120        reg1_in<0>
N116        N92         reg1_in<10>
N99         N91         reg1_in<11>
N82         N90         reg1_in<12>
N63         N89         reg1_in<13>
N194        N88         reg1_in<14>
N178        N2          reg1_in<15>
N101        N119        reg1_in<1>
N86         N118        reg1_in<2>
N68         N117        reg1_in<3>
N199        N116        reg1_in<4>
N182        N115        reg1_in<5>
N163        N114        reg1_in<6>
N147        N95         reg1_in<7>
N130        N94         reg1_in<8>
N113        N93         reg1_in<9>
N100        N136        reg2_in<0>
N172        N126        reg2_in<10>
N154        N125        reg2_in<11>
N137        N124        reg2_in<12>
N121        N123        reg2_in<13>
N104        N122        reg2_in<14>
N89         N121        reg2_in<15>
N84         N135        reg2_in<1>
N65         N134        reg2_in<2>
```

```
    N196        N133        reg2_in<3>
    N179        N132        reg2_in<4>
    N160        N131        reg2_in<5>
    N144        N130        reg2_in<6>
    N128        N129        reg2_in<7>
    N112        N128        reg2_in<8>
    N97         N127        reg2_in<9>
    N81         N113        reg3_in<0>
    N85         N45         reg3_in<10>
    N66         N39         reg3_in<11>
    N198        N38         reg3_in<12>
    N180        N37         reg3_in<13>
    N161        N36         reg3_in<14>
    N145        N137        reg3_in<15>
    N62         N112        reg3_in<1>
    N193        N111        reg3_in<2>
    N177        N110        reg3_in<3>
    N159        N109        reg3_in<4>
    N143        N108        reg3_in<5>
    N127        N107        reg3_in<6>
    N111        N48         reg3_in<7>
    N96         N47         reg3_in<8>
    N79         N46         reg3_in<9>
    N61         N26         reg4_in<0>
    N139        N15         reg4_in<10>
    N123        N10         reg4_in<11>
    N106        N9          reg4_in<12>
    N91         N8          reg4_in<13>
    N72         N7          reg4_in<14>
    N204        N6          reg4_in<15>
    N192        N25         reg4_in<1>
    N176        N23         reg4_in<2>
    N158        N22         reg4_in<3>
    N142        N21         reg4_in<4>
    N126        N20         reg4_in<5>
    N110        N19         reg4_in<6>
    N95         N18         reg4_in<7>
    N77         N17         reg4_in<8>
    N59         N16         reg4_in<9>
    N191        N3          reg5_in<0>
    N200        N14         reg5_in<10>
    N183        N54         reg5_in<11>
    N164        N53         reg5_in<12>
    N148        N52         reg5_in<13>
    N131        N50         reg5_in<14>
    N114        N55         reg5_in<15>
    N175        N4          reg5_in<1>
    N157        N40         reg5_in<2>
    N141        N41         reg5_in<3>
    N125        N42         reg5_in<4>
    N108        N43         reg5_in<5>
    N93         N44         reg5_in<6>
    N75         N11         reg5_in<7>
    N206        N12         reg5_in<8>
    N188        N13         reg5_in<9>
    N173        N86         reg6_in<0>
```

```
    N109        N76         reg6_in<10>
    N94         N75         reg6_in<11>
    N76         N74         reg6_in<12>
    N207        N73         reg6_in<13>
    N189        N72         reg6_in<14>
    N170        N71         reg6_in<15>
    N155        N85         reg6_in<1>
    N138        N84         reg6_in<2>
    N122        N83         reg6_in<3>
    N105        N82         reg6_in<4>
    N90         N81         reg6_in<5>
    N71         N80         reg6_in<6>
    N203        N79         reg6_in<7>
    N186        N78         reg6_in<8>
    N168        N77         reg6_in<9>
    N153        N51         reg7_in<0>
    N165        N33         reg7_in<10>
    N149        N32         reg7_in<11>
    N132        N31         reg7_in<12>
    N115        N30         reg7_in<13>
    N98         N29         reg7_in<14>
    N80         N87         reg7_in<15>
    N136        N28         reg7_in<1>
    N120        N143        reg7_in<2>
    N103        N142        reg7_in<3>
    N88         N141        reg7_in<4>
    N70         N140        reg7_in<5>
    N202        N139        reg7_in<6>
    N185        N49         reg7_in<7>
    N167        N35         reg7_in<8>
    N151        N34         reg7_in<9>
    N134        N70         reg8_in<0>
    N78         N60         reg8_in<10>
    N60         N59         reg8_in<11>
    N190        N58         reg8_in<12>
    N171        N57         reg8_in<13>
    N152        N56         reg8_in<14>
    N135        N102        reg8_in<15>
    N118        N69         reg8_in<1>
    N102        N68         reg8_in<2>
    N87         N67         reg8_in<3>
    N69         N66         reg8_in<4>
    N201        N65         reg8_in<5>
    N184        N64         reg8_in<6>
    N166        N63         reg8_in<7>
    N150        N62         reg8_in<8>
    N133        N61         reg8_in<9>
    N119        N0          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

```

```
                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   552     374
        total                    552     374

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   208     208
        total                    208     208

                                 terminals
        un-matched               0       0
        matched but
        different type           1       1
        total                    149     149


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

Code Listing 1.2: 8 to 1 MUX Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
         208              nets
         149              terminals
         276              pmos
         276              nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
         208              nets
         149              terminals
         187              pmos
         187              nmos


    Terminal correspondence points
    N74         N1          gnd!
    N138        N102        in<0>
    N147        N112        in<10>
    N130        N113        in<11>
    N113        N114        in<12>
    N97         N115        in<13>
    N79         N94         in<14>
    N61         N95         in<15>
    N121        N103        in<1>
    N104        N104        in<2>
    N88         N105        in<3>
    N69         N106        in<4>
    N201        N107        in<5>
    N185        N108        in<6>
    N169        N109        in<7>
    N154        N110        in<8>
    N136        N111        in<9>
    N84         N86         read_sel<0>
    N65         N87         read_sel<1>
    N196        N88         read_sel<2>
    N120        N30         reg1_in<0>
    N119        N66         reg1_in<10>
    N101        N67         reg1_in<11>
    N83         N68         reg1_in<12>
    N64         N69         reg1_in<13>
    N195        N70         reg1_in<14>
    N180        N71         reg1_in<15>
    N103        N31         reg1_in<1>
    N87         N32         reg1_in<2>
    N68         N33         reg1_in<3>
```

```
    N199        N34         reg1_in<4>
    N183        N97         reg1_in<5>
    N166        N98         reg1_in<6>
    N151        N99         reg1_in<7>
    N133        N10         reg1_in<8>
    N116        N65         reg1_in<9>
    N102        N2          reg2_in<0>
    N175        N137        reg2_in<10>
    N159        N136        reg2_in<11>
    N142        N131        reg2_in<12>
    N125        N144        reg2_in<13>
    N107        N143        reg2_in<14>
    N91         N142        reg2_in<15>
    N85         N3          reg2_in<1>
    N66         N4          reg2_in<2>
    N197        N5          reg2_in<3>
    N181        N6          reg2_in<4>
    N164        N156        reg2_in<5>
    N149        N155        reg2_in<6>
    N132        N140        reg2_in<7>
    N115        N139        reg2_in<8>
    N99         N138        reg2_in<9>
    N82         N152        reg3_in<0>
    N86         N36         reg3_in<10>
    N67         N37         reg3_in<11>
    N198        N154        reg3_in<12>
    N182        N130        reg3_in<13>
    N165        N129        reg3_in<14>
    N150        N128        reg3_in<15>
    N63         N151        reg3_in<1>
    N194        N149        reg3_in<2>
    N179        N148        reg3_in<3>
    N163        N147        reg3_in<4>
    N148        N146        reg3_in<5>
    N131        N145        reg3_in<6>
    N114        N96         reg3_in<7>
    N98         N150        reg3_in<8>
    N80         N35         reg3_in<9>
    N62         N46         reg4_in<0>
    N144        N24         reg4_in<10>
    N127        N25         reg4_in<11>
    N109        N26         reg4_in<12>
    N93         N27         reg4_in<13>
    N73         N28         reg4_in<14>
    N205        N29         reg4_in<15>
    N193        N47         reg4_in<1>
    N178        N48         reg4_in<2>
    N162        N49         reg4_in<3>
    N146        N38         reg4_in<4>
    N129        N75         reg4_in<5>
    N112        N76         reg4_in<6>
    N96         N77         reg4_in<7>
    N77         N22         reg4_in<8>
    N59         N23         reg4_in<9>
    N192        N127        reg5_in<0>
    N200        N40         reg5_in<10>
```

```
111     N184        N41         reg5_in<11>
112     N167        N42         reg5_in<12>
113     N152        N43         reg5_in<13>
114     N134        N44         reg5_in<14>
115     N117        N45         reg5_in<15>
116     N177        N126        reg5_in<1>
117     N161        N125        reg5_in<2>
118     N145        N153        reg5_in<3>
119     N128        N11         reg5_in<4>
120     N110        N12         reg5_in<5>
121     N94         N13         reg5_in<6>
122     N75         N116        reg5_in<7>
123     N206        N117        reg5_in<8>
124     N189        N39         reg5_in<9>
125     N176        N83         reg6_in<0>
126     N111        N119        reg6_in<10>
127     N95         N120        reg6_in<11>
128     N76         N121        reg6_in<12>
129     N207        N122        reg6_in<13>
130     N190        N123        reg6_in<14>
131     N173        N124        reg6_in<15>
132     N160        N84         reg6_in<1>
133     N143        N85         reg6_in<2>
134     N126        N59         reg6_in<3>
135     N108        N60         reg6_in<4>
136     N92         N61         reg6_in<5>
137     N72         N62         reg6_in<6>
138     N204        N63         reg6_in<7>
139     N188        N64         reg6_in<8>
140     N172        N118        reg6_in<9>
141     N158        N89         reg7_in<0>
142     N168        N80         reg7_in<10>
143     N153        N81         reg7_in<11>
144     N135        N82         reg7_in<12>
145     N118        N72         reg7_in<13>
146     N100        N73         reg7_in<14>
147     N81         N74         reg7_in<15>
148     N141        N90         reg7_in<1>
149     N124        N91         reg7_in<2>
150     N106        N92         reg7_in<3>
151     N90         N93         reg7_in<4>
152     N71         N7          reg7_in<5>
153     N203        N8          reg7_in<6>
154     N187        N9          reg7_in<7>
155     N171        N78         reg7_in<8>
156     N156        N79         reg7_in<9>
157     N139        N14         reg8_in<0>
158     N78         N53         reg8_in<10>
159     N60         N54         reg8_in<11>
160     N191        N55         reg8_in<12>
161     N174        N56         reg8_in<13>
162     N157        N57         reg8_in<14>
163     N140        N58         reg8_in<15>
164     N123        N15         reg8_in<1>
165     N105        N16         reg8_in<2>
166     N89         N17         reg8_in<3>
```

```
    N70         N18         reg8_in<4>
    N202        N19         reg8_in<5>
    N186        N20         reg8_in<6>
    N170        N21         reg8_in<7>
    N155        N51         reg8_in<8>
    N137        N52         reg8_in<9>
    N122        N0          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   552     374
        total                    552     374

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   208     208
        total                    208     208

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    149     149


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout
```

```
223
224 devbad.out:
225
226 netbad.out:
227
228 mergenet.out:
229
230 termbad.out:
231
232 prunenet.out:
233
234 prunedev.out:
235
236 audit.out:
```

Code Listing 1.3: 1 to 8 DEMUX LVS Result

```
1  @(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $
2
3  Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
       home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
       cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
4  Like matching is enabled.
5  Net swapping is enabled.
6  Using terminal names as correspondence points.
7  Compiling Diva LVS rules...
8
9      Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
10        count
11         40              nets
12         13              terminals
13         68              pmos
14         68              nmos
15
16     Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
17        count
18         40              nets
19         13              terminals
20         35              pmos
21         35              nmos
22
23
24     Terminal correspondence points
25     N28         N1          gnd!
26     N35         N13         in<0>
27     N33         N12         in<1>
28     N31         N11         in<2>
29     N32         N16         out<0>
30     N30         N18         out<1>
31     N29         N19         out<2>
32     N27         N20         out<3>
33     N39         N22         out<4>
34     N38         N23         out<5>
35     N37         N15         out<6>
36     N36         N21         out<7>
37     N34         N0          vdd!
```

```
38
39 Devices in the netlist but not in the rules:
40         pcapacitor
41 Devices in the rules but not in the netlist:
42         cap nfet pfet nmos4 pmos4
43
44 The net-lists match.
45
46                              layout  schematic
47                                 instances
48         un-matched              0       0
49         rewired                 0       0
50         size errors             0       0
51         pruned                  0       0
52         active                  136     70
53         total                   136     70
54
55                                   nets
56         un-matched              0       0
57         merged                  0       0
58         pruned                  0       0
59         active                  40      40
60         total                   40      40
61
62                                 terminals
63         un-matched              0       0
64         matched but
65         different type          0       0
66         total                   13      13
67
68
69 Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
70
71 devbad.out:
72
73 netbad.out:
74
75 mergenet.out:
76
77 termbad.out:
78
79 prunenet.out:
80
81 prunedev.out:
82
83 audit.out:
84
85
86 Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout
87
88 devbad.out:
89
90 netbad.out:
91
92 mergenet.out:
93
```

```
94 termbad.out:
95
96 prunenet.out:
97
98 prunedev.out:
99
100 audit.out:
```

Code Listing 1.4: 3 to 8 Decoder LVS Result

## 1.2 Execution Stage

The Execution (EX) stage will do the arithmetic operations and store the final result in Register file or Memory bank. For this final project, our EX stage architecture support 17 kinds of instructions, including STORE, LOAD, AND etc,. For the MUL/MULI commands, we use Out-of-Order instruction by putting it outside the arithmetic logic unit (ALU). Thus, we could realize parallelism of the MUL/MULI instructions. For the AND/OR logic blocks, we use dynamic logic to reduce dynamic power consumption. Moreover, our adder support both add and subtract operations. The following part will show the schematic, layout, functionality result and optimization of our design.

### 1.2.1 5 bit Multiplier

Figure 1.5 shows the schematics of the MUL block. And since we only need do 5-bit and 5-bit multiplication, we set the most significant bit of the two inputs the same as the most significant bit of the input. And also since we need a 16-bit output data, we add 6 more bit to the left and make it the same as the most significant bit of our multiplier's output.

#### Schematic Design

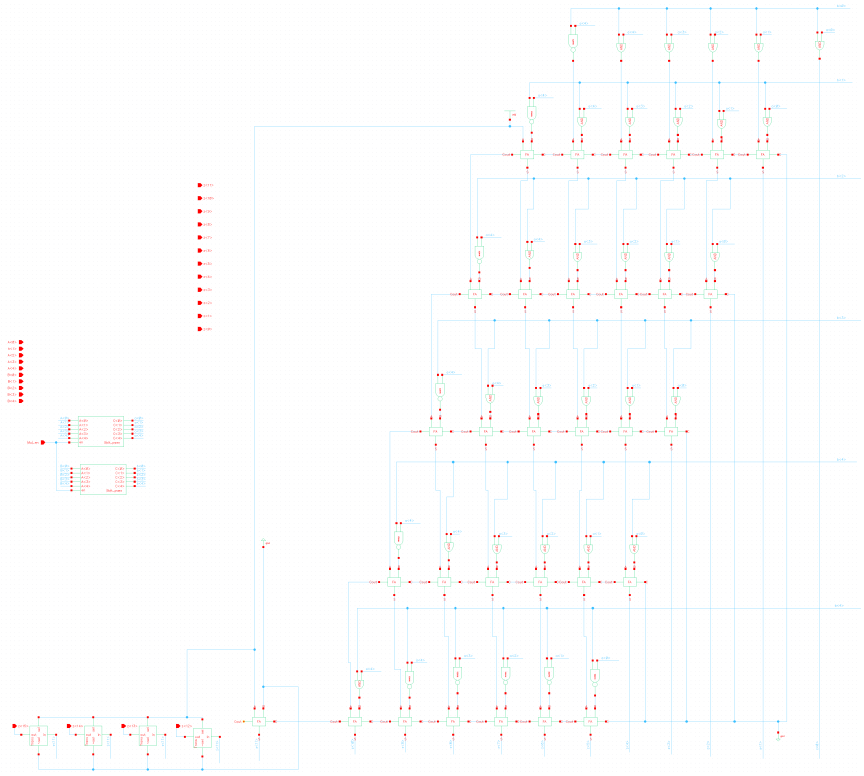

**Figure 1.5:** *5bit Multiplier schematic*

### Layout Design

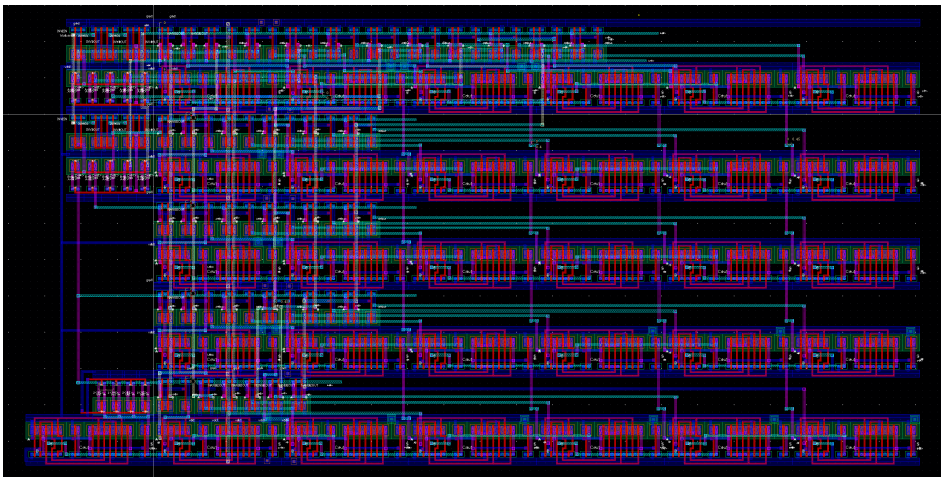

**Figure 1.6:** *5bit Multiplier Layout*

### Block Simulation

Figure 1.7 shows the simulation for the multiplication block in the ALU. Input A¡0:4¿=0b00011 and B¡0:4¿=0b00110, the expected output should be 3*6=18=0b010010 and all the upper bits are 0, the simulation is as expected.

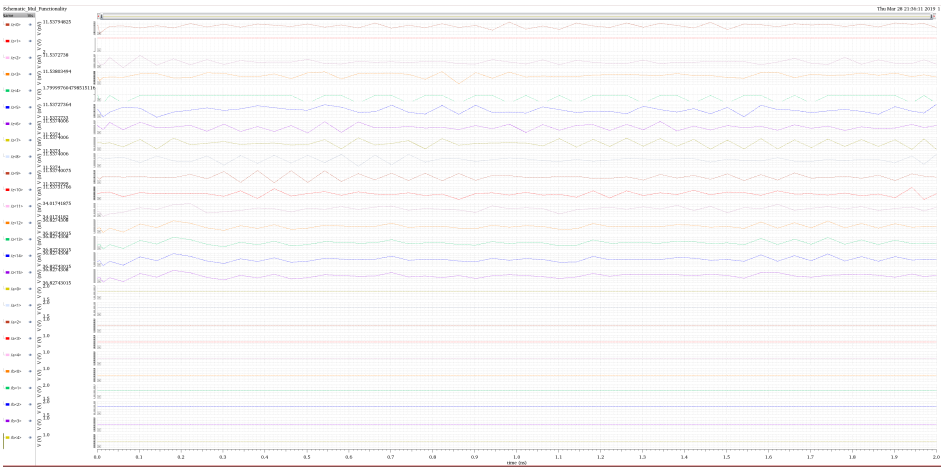

**Figure 1.7:** *5bit Multiplier Simulation*

### LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
         565             nets
         29              terminals
         566             pmos
         566             nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
         565             nets
         29              terminals
         552             pmos
         552             nmos


    Terminal correspondence points
    N555        N109        A<0>
    N551        N88         A<1>
    N547        N91         A<2>
    N542        N98         A<3>
    N567        N108        A<4>
    N570        N99         B<0>
    N566        N101        B<1>
    N563        N102        B<2>
    N560        N92         B<3>
    N557        N112        B<4>
    N558        N122        Mul_en
    N545        N0          gnd!
    N554        N1          vdd!
    N548        N75         z<0>
    N552        N87         z<10>
    N549        N119        z<11>
    N543        N120        z<12>
    N568        N115        z<13>
    N564        N104        z<14>
    N561        N123        z<15>
    N544        N65         z<1>
    N569        N86         z<2>
    N565        N79         z<3>
    N562        N82         z<4>
    N559        N72         z<5>
    N556        N77         z<6>
    N553        N70         z<7>
    N550        N71         z<8>
```

```
    N546        N68         z<9>

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   1132    1104
        total                    1132    1104

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   565     565
        total                    565     565

                                 terminals
        un-matched               0       0
        matched but
        different type           4       4
        total                    29      29


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout

devbad.out:

netbad.out:

mergenet.out:
```

```
109
110 termbad.out:
111
112 prunenet.out:
113
114 prunedev.out:
115
116 audit.out:
```

Code Listing 1.5: 5 bit Multiplier LVS Result

### 1.2.2 Shifter

For Bit Shift Block, to realize the bit shift function, we applied the structure in Figure 1.8. There is a 2-to-1 MUX to decide whether it's left shifting or right shifting.[3]

#### Schematic Design

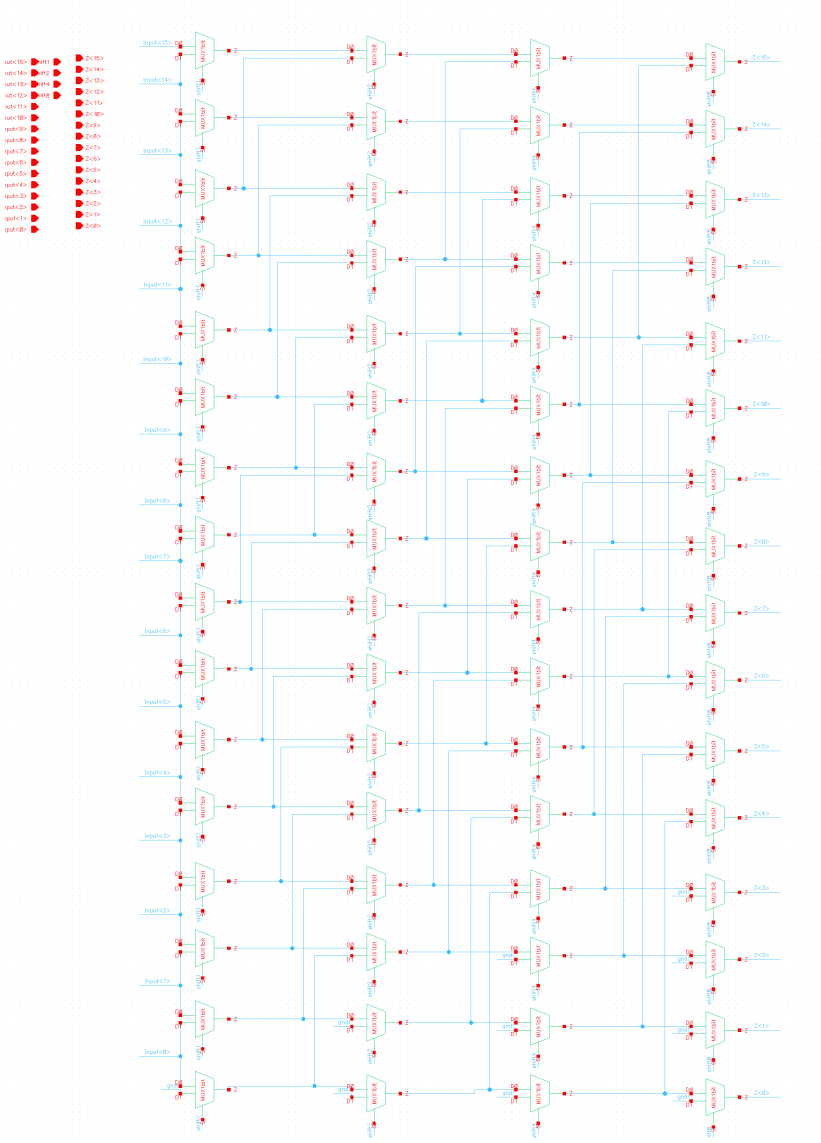

**Figure 1.8:** *Shifter schematic*

### Layout

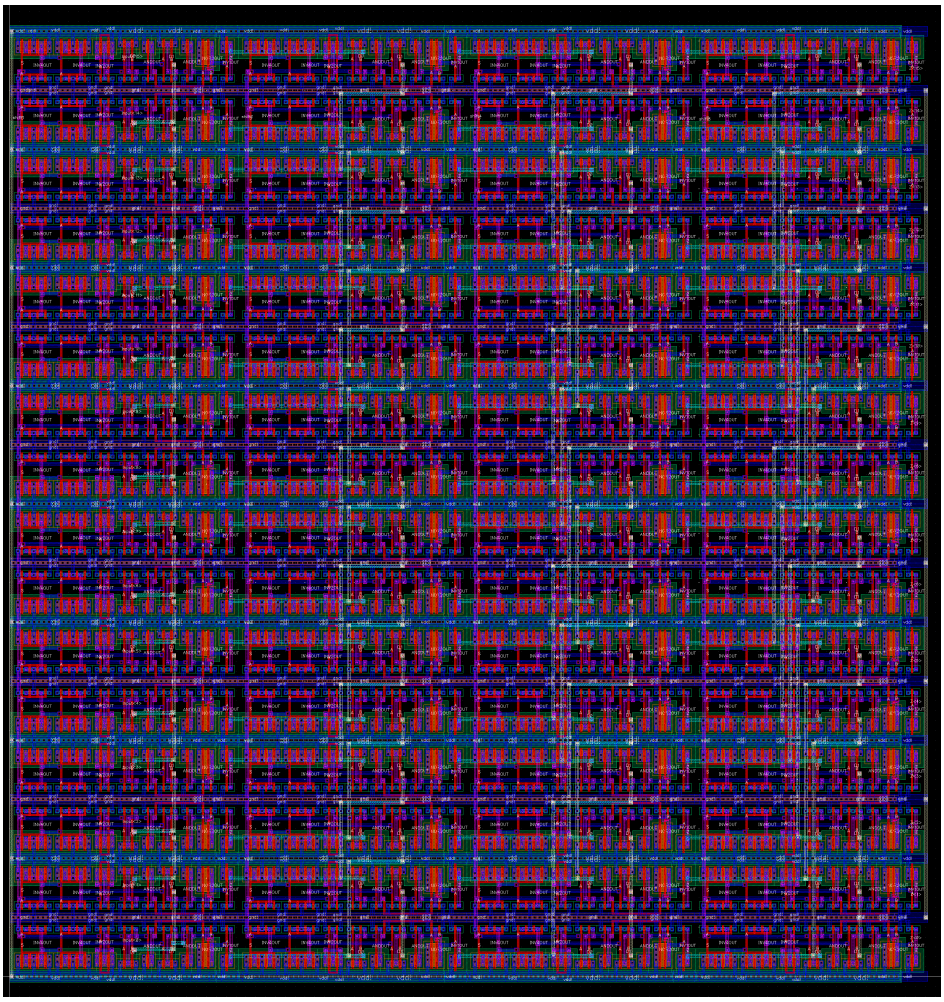

**Figure 1.9:** *Shifter Layout*

### Block Simulation

Figure 1.10 shows the simulation for the shift right logic block in the ALU. The shift signal=0b0001, and the input A¡0:15¿=0b1100, we wish to shift input 1 bit to the left, the output collected was shift ¡0:15¿=0b11000, which is as expected.

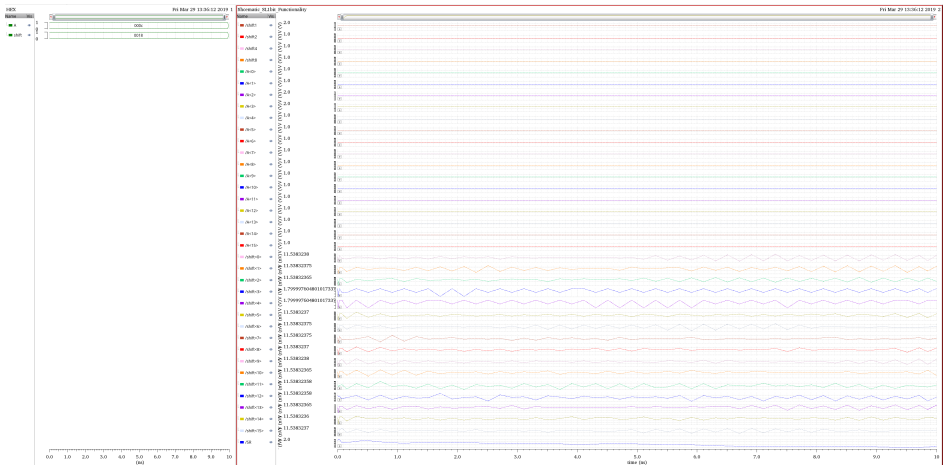

**Figure 1.10:** *Shift Left Simulation*

Figure 1.11 shows the simulation for the shift right logic block in the ALU. The shift signal=0b0001, and the input A¡0:15¿=0b1100, we wish to shift input 1 bit to the right, the output collected was shift ¡0:15¿=0b0110, which is as expected

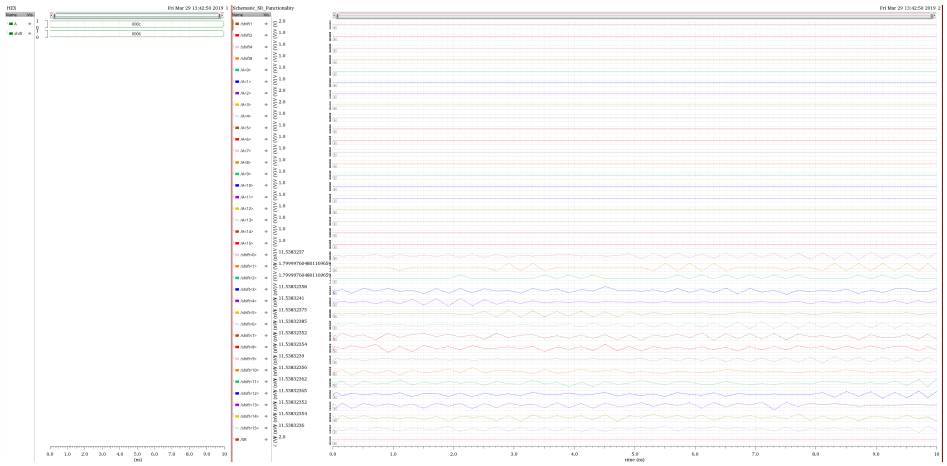

**Figure 1.11:** *Shift Right Simulation*

### LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
        790             nets
        38              terminals
        1216            pmos
        1216            nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
        790             nets
        38              terminals
        768             pmos
        768             nmos


    Terminal correspondence points
    N779        N23         Input<0>
    N776        N33         Input<10>
    N766        N34         Input<11>
    N761        N35         Input<12>
    N757        N36         Input<13>
    N752        N37         Input<14>
    N786        N38         Input<15>
    N775        N24         Input<1>
    N765        N25         Input<2>
    N760        N26         Input<3>
    N755        N27         Input<4>
    N789        N28         Input<5>
    N785        N29         Input<6>
    N782        N30         Input<7>
    N780        N31         Input<8>
    N777        N32         Input<9>
    N768        N43         Z<0>
    N770        N14         Z<10>
    N763        N13         Z<11>
    N759        N12         Z<12>
    N754        N11         Z<13>
    N788        N8          Z<14>
    N784        N7          Z<15>
    N762        N42         Z<1>
    N758        N41         Z<2>
    N753        N40         Z<3>
    N787        N39         Z<4>
    N783        N22         Z<5>
```

```
    N781        N21         Z<6>
    N778        N20         Z<7>
    N774        N19         Z<8>
    N764        N18         Z<9>
    N756        N0          gnd!
    N772        N6          shift1
    N771        N15         shift2
    N769        N16         shift4
    N767        N17         shift8
    N773        N1          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   2432    1536
        total                    2432    1536

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   790     790
        total                    790     790

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    38      38


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

```
109
110
111 Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout
112
113 devbad.out:
114
115 netbad.out:
116
117 mergenet.out:
118
119 termbad.out:
120
121 prunenet.out:
122
123 prunedev.out:
124
125 audit.out:
```

Code Listing 1.6: Shifter LVS Result

### 1.2.3 Adder

Figure 1.12 shows the schematics of the ADD and MIN block. The ADD part is directly connecting 16 Full-Adder together, and the MIN part is also based on this design.

#### Schematic Design

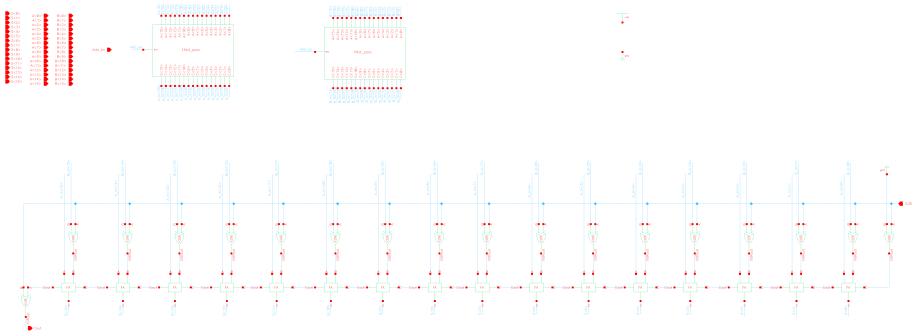

**Figure 1.12:** *Adder schematic*

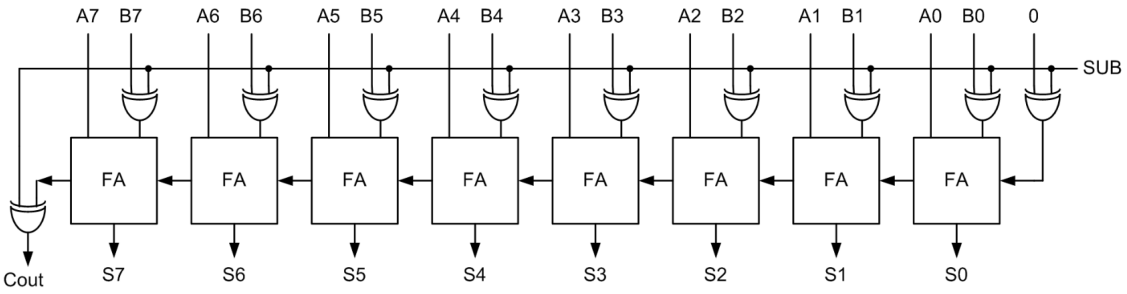


**Figure 1.13:** *Concept of Change an Adder to a Subtractor[1]*

For MIN function, we use the subtractor to decide which value is smaller. For example, we do subtration of the two values and see if the most significant bit is "0" or "1", if it's "0", then A-B is a positive value, which means A is larger than B, and then the MIN output should be B and vice versa. Figure 1.13 shows the method to change an adder to a subtractor, which is by adding XOR gates. And we use a 2-to-1 MUX to decide which value we should choose.

### Layout Design

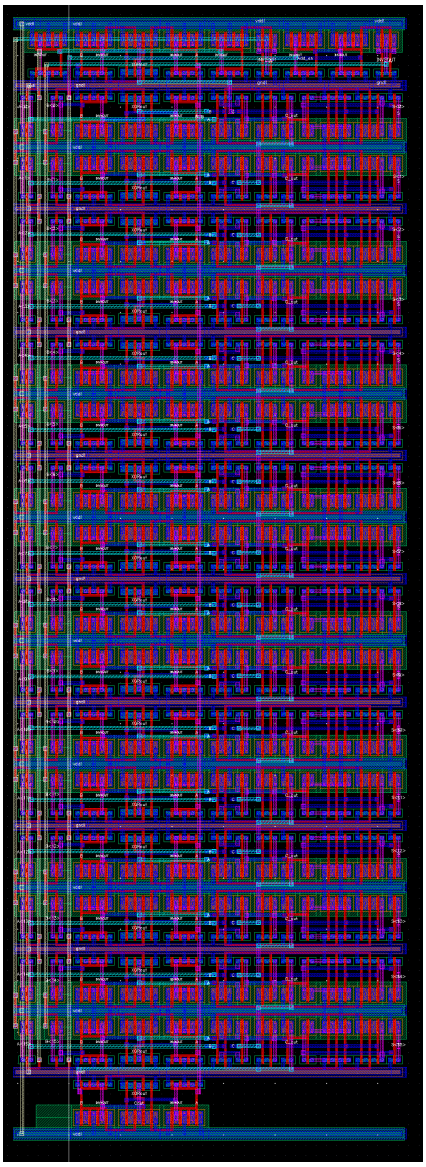

**Figure 1.14:** *Adder Layout*

### Block Simulation

Figure 1.15 shows the simulation result of the ADD block functionality. The input A¡0:15¿ is 0h0181 and input B¡0:15¿ is 0h92FF, then the output is 0h9480.And in Figure 1.16 Since the value of B A is larger than B, then the output should be B, which is 0h0181, and the output in simulation is as expected.

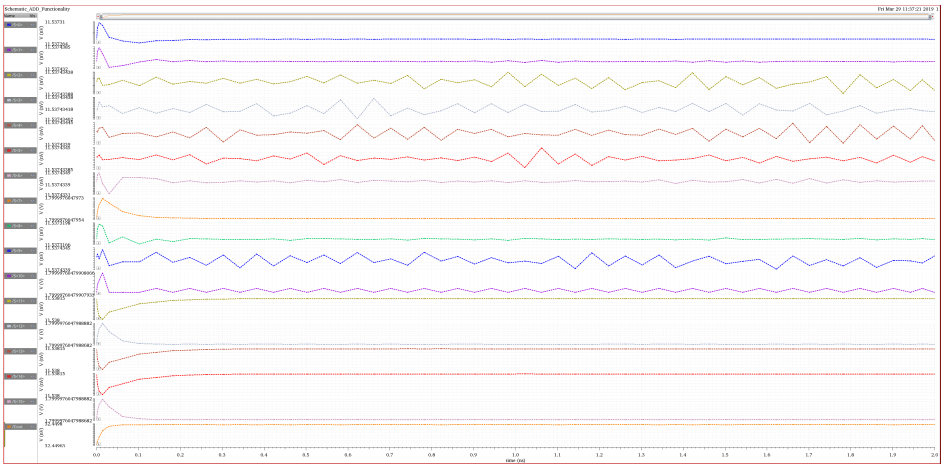

**Figure 1.15:** *Adder Simulation*

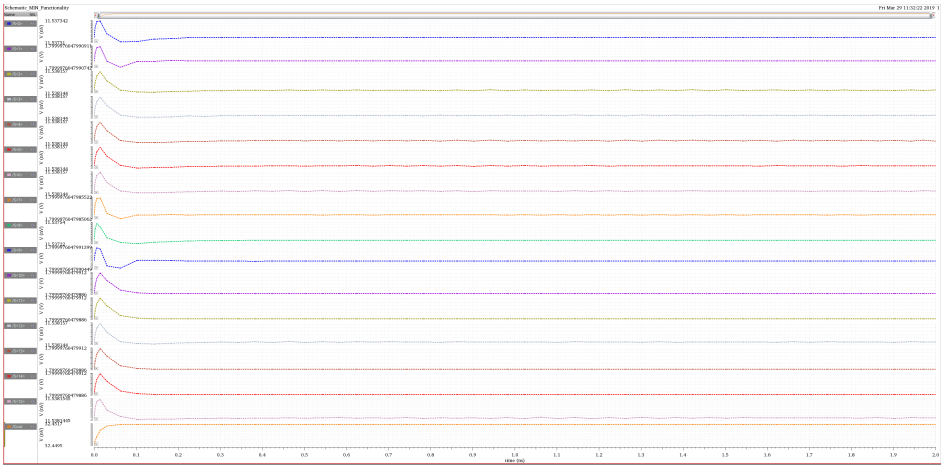

**Figure 1.16:** *Min Block Simulation*

### LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
```

```
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
         424              nets
         53               terminals
         492              pmos
         492              nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
         424              nets
         53               terminals
         370              pmos
         370              nmos


    Terminal correspondence points
    N394        N83         A<0>
    N381        N69         A<10>
    N374        N68         A<11>
    N421        N67         A<12>
    N414        N40         A<13>
    N410        N33         A<14>
    N405        N32         A<15>
    N387        N78         A<1>
    N380        N77         A<2>
    N373        N76         A<3>
    N420        N75         A<4>
    N413        N74         A<5>
    N409        N73         A<6>
    N404        N72         A<7>
    N400        N71         A<8>
    N391        N70         A<9>
    N384        N6          Add_en
    N422        N31         B<0>
    N397        N19         B<10>
    N389        N18         B<11>
    N383        N17         B<12>
    N376        N16         B<13>
    N423        N15         B<14>
    N417        N4          B<15>
    N416        N30         B<1>
    N411        N27         B<2>
    N406        N26         B<3>
    N401        N25         B<4>
    N392        N24         B<5>
    N386        N23         B<6>
    N379        N22         B<7>
    N372        N21         B<8>
    N419        N20         B<9>
    N415        N119        Cout
    N398        N7          S<0>
    N408        N36         S<10>
```

```
    N403        N37         S<11>
    N396        N81         S<12>
    N388        N104        S<13>
    N382        N29         S<14>
    N375        N3          S<15>
    N390        N8          S<1>
    N385        N9          S<2>
    N378        N10         S<3>
    N371        N11         S<4>
    N418        N12         S<5>
    N412        N13         S<6>
    N407        N14         S<7>
    N402        N34         S<8>
    N395        N35         S<9>
    N399        N2          SUB
    N377        N1          gnd!
    N393        N0          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   984     740
        total                    984     740

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   424     424
        total                    424     424

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    53      53


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

```

```
117 termbad.out:
118
119 prunenet.out:
120
121 prunedev.out:
122
123 audit.out:
124
125
126 Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout
127
128 devbad.out:
129
130 netbad.out:
131
132 mergenet.out:
133
134 termbad.out:
135
136 prunenet.out:
137
138 prunedev.out:
139
140 audit.out:
```

Code Listing 1.7: Adder LVS Result

### 1.2.4 And Logic

#### Schematic Design

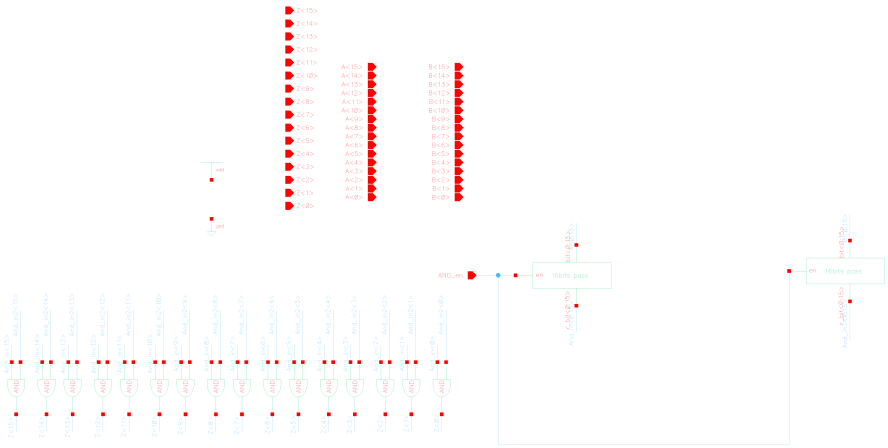

**Figure 1.17:** *And schematic*

#### Layout Design

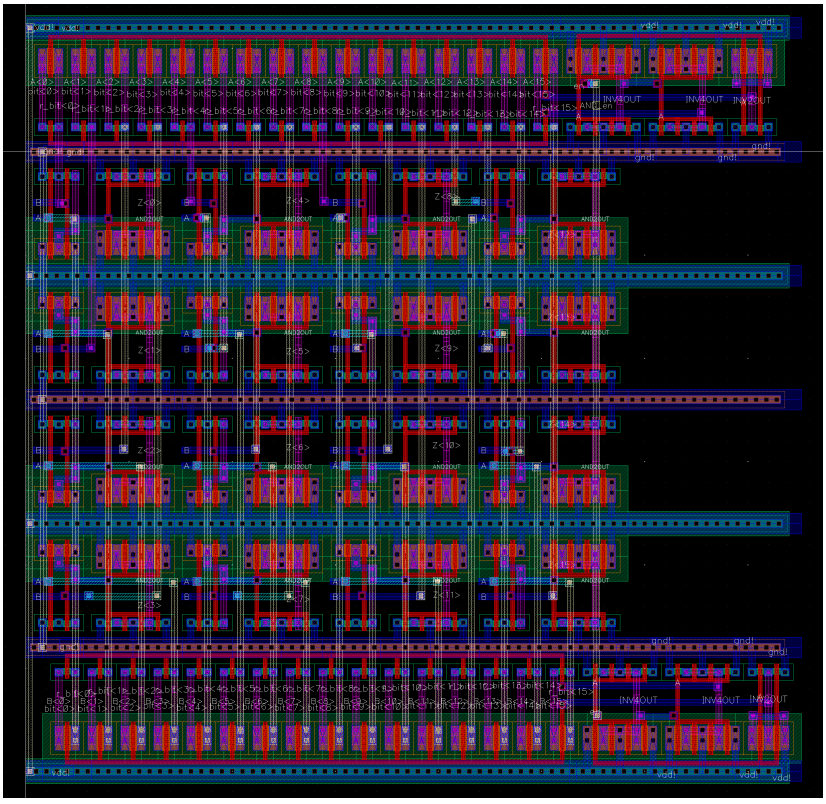

**Figure 1.18:** *And Layout*

### Block Simulation

Figure 1.19 shows the output simulation for the AND logic gate. For the benefit of simplicity, we only shows 2-bit input A¡0:1¿ and B¡0:1¿. Our input A¡0:1¿=01 and B¡0:1¿=11, the output Z¡0:1¿=A(and)B shown is 01, which is as expected.

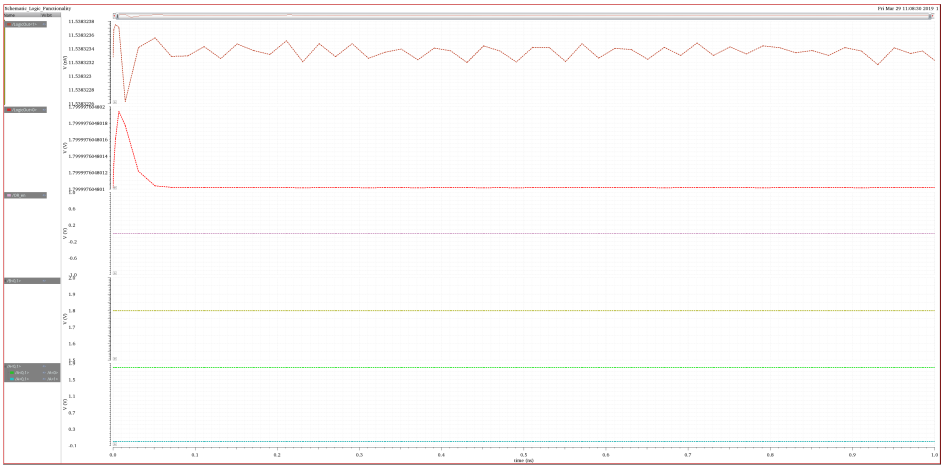

**Figure 1.19:** *And Simulation*

### LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
         121             nets
         51              terminals
         148             pmos
         148             nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
         121             nets
         51              terminals
         86              pmos
         86              nmos


    Terminal correspondence points
    N94         N58         A<0>
```

```
    N81         N48         A<10>
    N74         N47         A<11>
    N118        N46         A<12>
    N110        N45         A<13>
    N105        N44         A<14>
    N101        N43         A<15>
    N86         N57         A<1>
    N80         N56         A<2>
    N73         N55         A<3>
    N117        N54         A<4>
    N109        N53         A<5>
    N104        N52         A<6>
    N100        N51         A<7>
    N97         N50         A<8>
    N90         N49         A<9>
    N115        N26         AND_en
    N119        N81         B<0>
    N95         N64         B<10>
    N87         N63         B<11>
    N82         N62         B<12>
    N75         N61         B<13>
    N120        N60         B<14>
    N112        N59         B<15>
    N111        N79         B<1>
    N106        N78         B<2>
    N102        N77         B<3>
    N98         N76         B<4>
    N92         N75         B<5>
    N85         N74         B<6>
    N79         N72         B<7>
    N72         N71         B<8>
    N116        N70         B<9>
    N89         N41         Z<0>
    N91         N32         Z<10>
    N84         N31         Z<11>
    N78         N34         Z<12>
    N71         N33         Z<13>
    N114        N38         Z<14>
    N108        N37         Z<15>
    N83         N28         Z<1>
    N77         N27         Z<2>
    N70         N30         Z<3>
    N113        N29         Z<4>
    N107        N36         Z<5>
    N103        N35         Z<6>
    N99         N42         Z<7>
    N96         N39         Z<8>
    N88         N40         Z<9>
    N76         N0          gnd!
    N93         N1          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

```

```
The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   296     172
        total                    296     172

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   121     121
        total                    121     121

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    51      51


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

```

```
138 audit.out:
```

Code Listing 1.8: And LVS Result

### Block Optimization

In the section below we implemented dynamic logic for AND, which helps reduce the number of transistors used compared to the static CMOS circuit. In addition, the static power loss is less in dynamic logic circuit, and switching speed is faster because of lower load capacitance $C_{load}$ and interconnection capacitance $C_{int}$

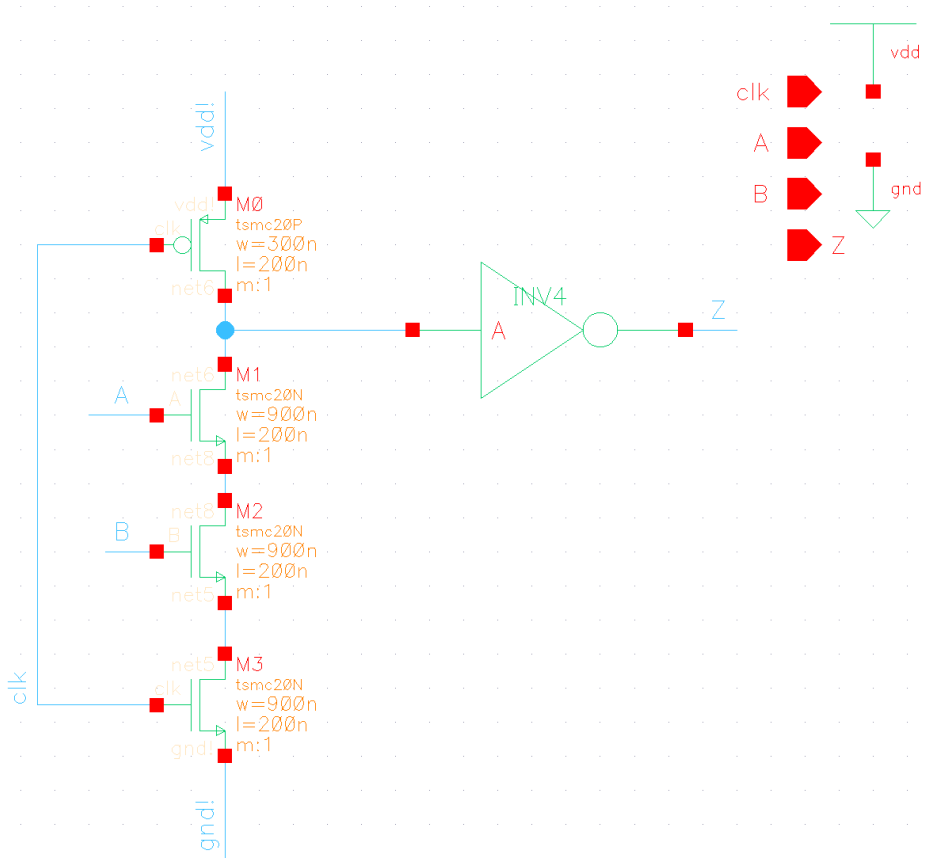


**Figure 1.20:** *And Logic Cell Optimization*

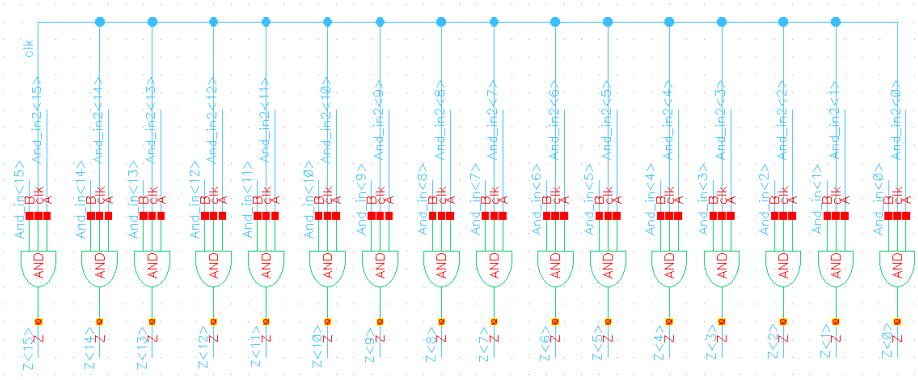


**Figure 1.21:** *And logic Optimization*

### 1.2.5 Or Logic

#### Schematic Design

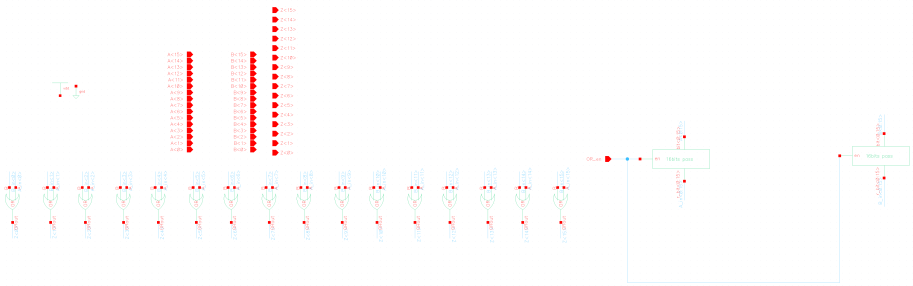

**Figure 1.22:** *OR schematic*

### Layout Design

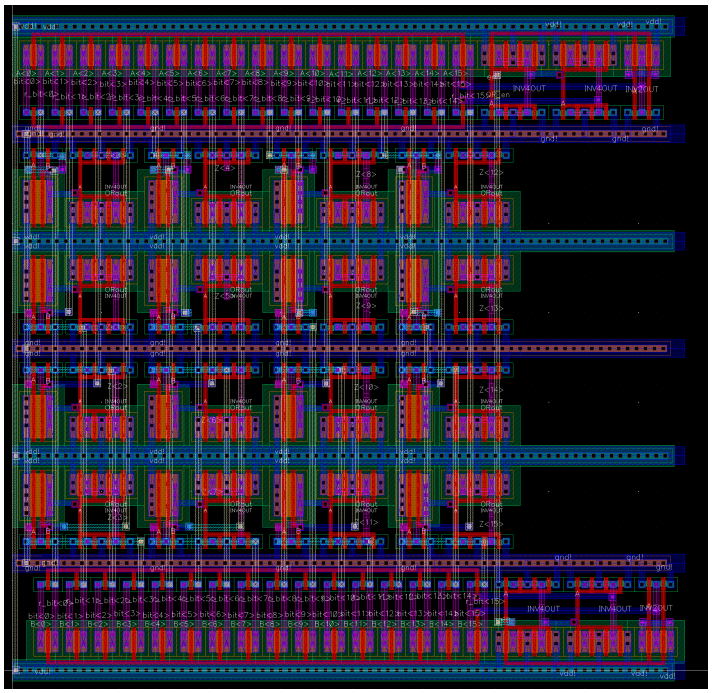

**Figure 1.23:** *Or Layout*

### Block Simulation

Figure 1.24 shows the output simulation for the OR logic gate. For the benefit of simplicity, again we only shows 2-bit input A¡0:1¿ and B¡0:1¿. Our input A¡0:1¿=01 and B¡0:1¿=11, the output Z¡0:1¿=A(or)B shown is 11, which is as expected.

### LVS Result

```
1  @(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $
2
3  Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
       home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
       cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
4  Like matching is enabled.
5  Net swapping is enabled.
6  Using terminal names as correspondence points.
7  Compiling Diva LVS rules...
8
9      Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
10        count
11          121             nets
12          51              terminals
13          148             pmos
14          148             nmos
```

**Figure 1.24:** *Or Simulation*

```
15
16    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
17        count
18          121               nets
19          51                terminals
20          86                pmos
21          86                nmos
22
23
24    Terminal correspondence points
25    N95         N64         A<0>
26    N82         N54         A<10>
27    N75         N53         A<11>
28    N119        N52         A<12>
29    N112        N51         A<13>
30    N107        N50         A<14>
31    N103        N49         A<15>
32    N87         N63         A<1>
33    N81         N62         A<2>
34    N74         N61         A<3>
35    N118        N60         A<4>
36    N111        N59         A<5>
37    N106        N58         A<6>
38    N102        N57         A<7>
39    N98         N56         A<8>
40    N91         N55         A<9>
41    N120        N15         B<0>
42    N96         N5          B<10>
43    N88         N4          B<11>
44    N83         N3          B<12>
45    N76         N2          B<13>
46    N121        N83         B<14>
47    N114        N82         B<15>
48    N113        N14         B<1>
49    N108        N13         B<2>
50    N104        N12         B<3>
51    N99         N11         B<4>
```

```
    N93         N10         B<5>
    N86         N9          B<6>
    N80         N8          B<7>
    N73         N7          B<8>
    N117        N6          B<9>
    N101        N32         OR_en
    N90         N34         Z<0>
    N92         N39         Z<10>
    N85         N40         Z<11>
    N79         N41         Z<12>
    N72         N42         Z<13>
    N116        N43         Z<14>
    N110        N47         Z<15>
    N84         N45         Z<1>
    N78         N33         Z<2>
    N71         N48         Z<3>
    N115        N46         Z<4>
    N109        N44         Z<5>
    N105        N35         Z<6>
    N100        N36         Z<7>
    N97         N37         Z<8>
    N89         N38         Z<9>
    N77         N0          gnd!
    N94         N1          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   296     172
        total                    296     172

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   121     121
        total                    121     121

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    51      51


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
```

```
108
109 devbad.out:
110
111 netbad.out:
112
113 mergenet.out:
114
115 termbad.out:
116
117 prunenet.out:
118
119 prunedev.out:
120
121 audit.out:
122
123
124 Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout
125
126 devbad.out:
127
128 netbad.out:
129
130 mergenet.out:
131
132 termbad.out:
133
134 prunenet.out:
135
136 prunedev.out:
137
138 audit.out:
```

Code Listing 1.9: Or LVS Result

### Block Optimization

In the section below we implemented dynamic logic for OR, which helps reduce the number of transistors used compared to the static CMOS circuit. In addition, the static power loss is less in dynamic logic circuit, and switching speed is faster because of lower load capacitance $C_{load}$ and interconnection capacitance $C_{int}$

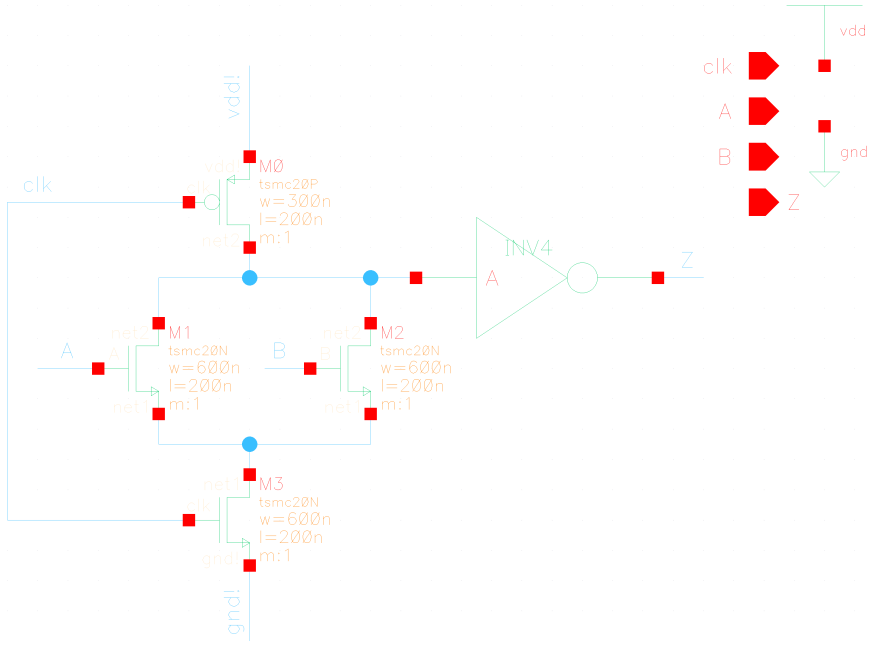


**Figure 1.25:** *Or Cell Optimization*

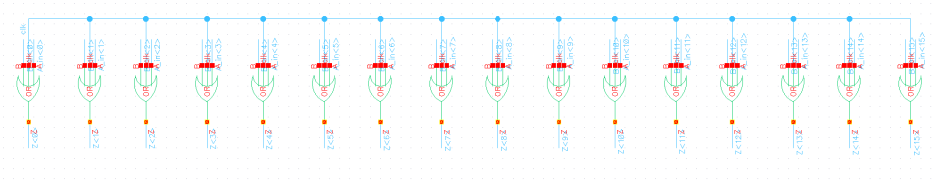

**Figure 1.26:** *Or logic Optimization*

## 1.3 Memory Stage

For this project we will only need 512 bit SRAM using two 256 memory banks, therefore we modified the previous design to fit this purpose and reduce the layout area.

### 1.3.1 Schematic Design

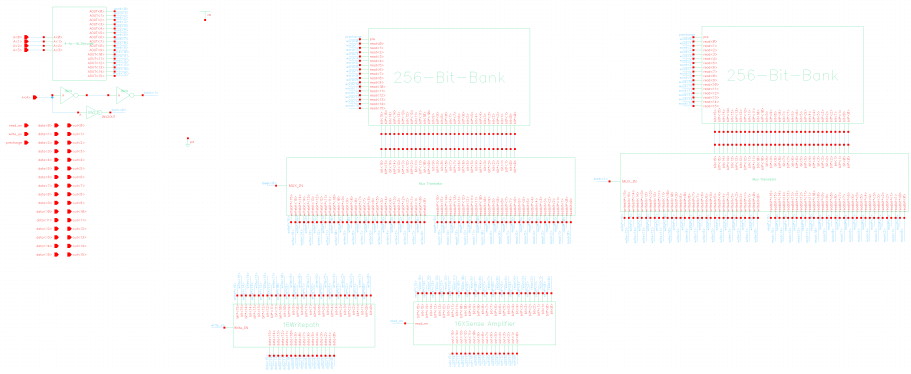


**Figure 1.27:** *Memory schematic*

### 1.3.2 Layout Design

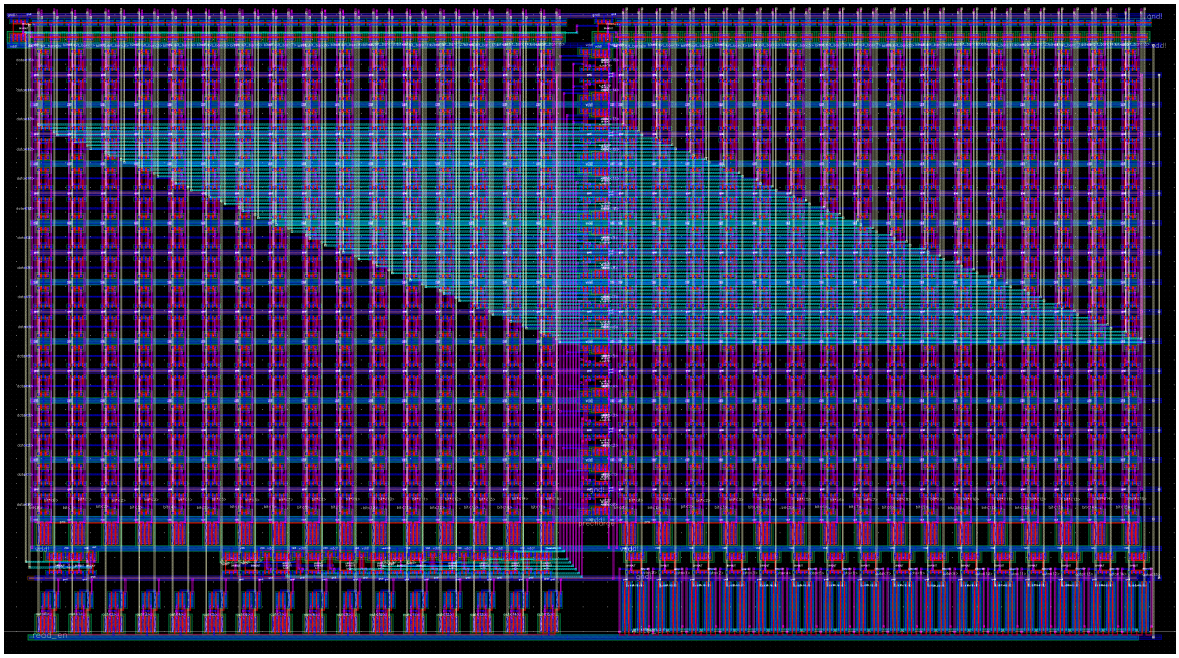

**Figure 1.28:** *Memory layout*

### 1.3.3 LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
        1355            nets
        42              terminals
        1490            pmos
        2466            nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
        1355            nets
        42              terminals
        1345            pmos
        2321            nmos


    Terminal correspondence points
    N1362     N126      A<0>
    N1355     N190      A<1>
    N1352     N40       A<2>
    N1346     N188      A<3>
    N1382     N189      A<4>
    N1347     N85       data<0>
    N1385     N75       data<10>
    N1380     N74       data<11>
    N1376     N73       data<12>
    N1372     N57       data<13>
    N1367     N37       data<14>
    N1361     N34       data<15>
    N1384     N84       data<1>
    N1379     N83       data<2>
    N1375     N82       data<3>
    N1371     N81       data<4>
    N1366     N80       data<5>
    N1359     N79       data<6>
    N1353     N78       data<7>
    N1350     N77       data<8>
    N1344     N76       data<9>
    N1349     N0        gnd!
    N1360     N102      out<0>
    N1383     N92       out<10>
    N1378     N91       out<11>
    N1374     N90       out<12>
    N1370     N89       out<13>
    N1365     N88       out<14>
```

```
    N1358       N87         out<15>
    N1354       N101        out<1>
    N1351       N100        out<2>
    N1345       N99         out<3>
    N1381       N98         out<4>
    N1377       N97         out<5>
    N1373       N96         out<6>
    N1369       N95         out<7>
    N1364       N94         out<8>
    N1357       N93         out<9>
    N1368       N185        precharge
    N1356       N86         read_en
    N1363       N1          vdd!
    N1348       N33         write_en

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   3956    3666
        total                    3956    3666

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   1355    1355
        total                    1355    1355

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    42      42


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:
```

```
109
110 prunedev.out:
111
112 audit.out:
113
114
115 Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout
116
117 devbad.out:
118
119 netbad.out:
120
121 mergenet.out:
122
123 termbad.out:
124
125 prunenet.out:
126
127 prunedev.out:
128
129 audit.out:
```

Code Listing 1.10: Memory LVS Result

## 1.4 Stage Registers

The section below show the schematic and layout of the stage registers. Namely, the stages are ID+EX, EX+MEM, and MEM+WB stages. The write back path is partially combined with the MEM+WB stage register for correct data arrival timing for correct functionality and partially combined with the front end, namely using the 1 to 8 demux to feed the data back to the register files.

### 1.4.1 Schematic Design

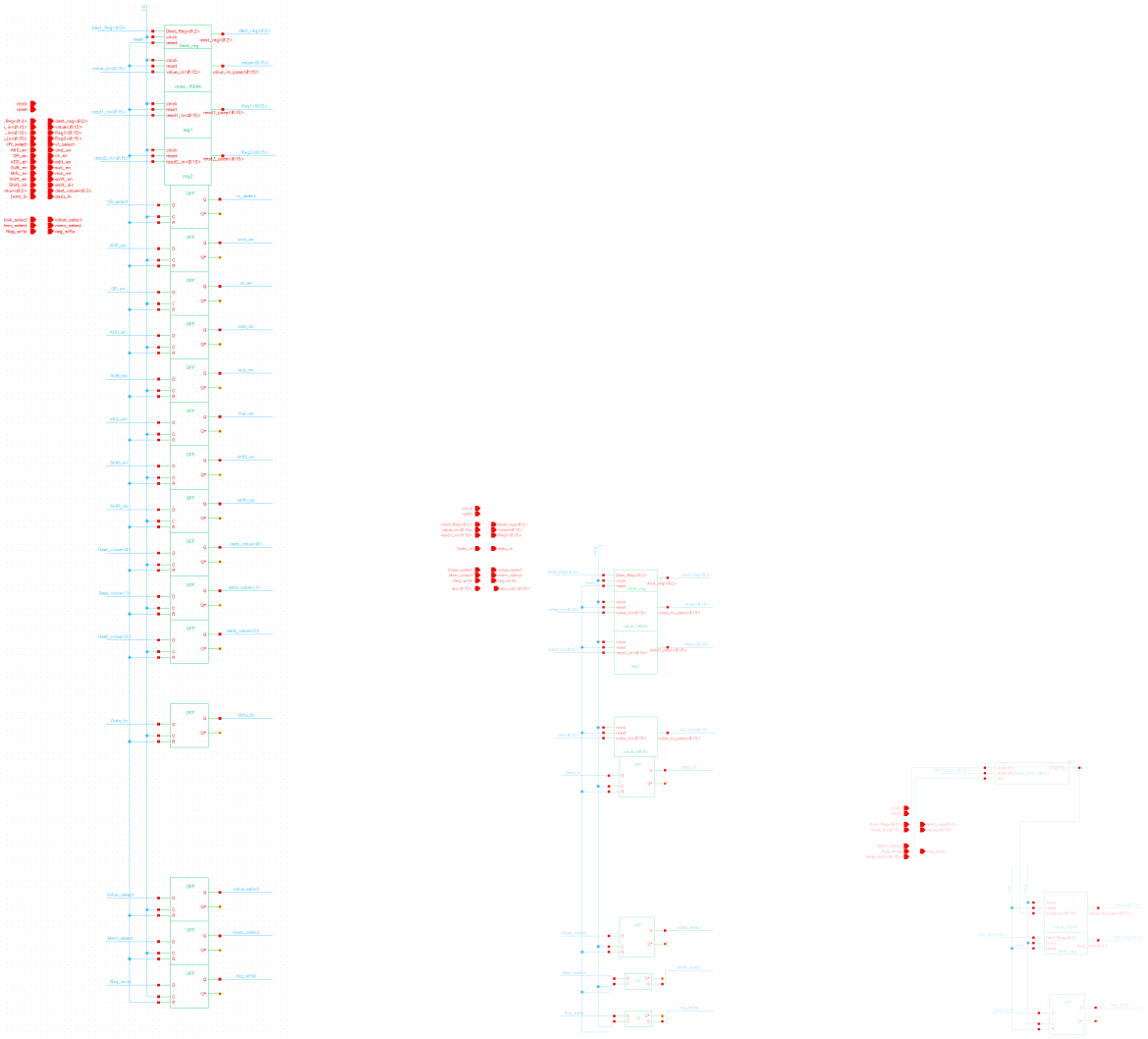

**Figure 1.29:** *Stage Register of ID+EX, EX+MEM, and MEM+WB schematic.*

### 1.4.2 Layout Design

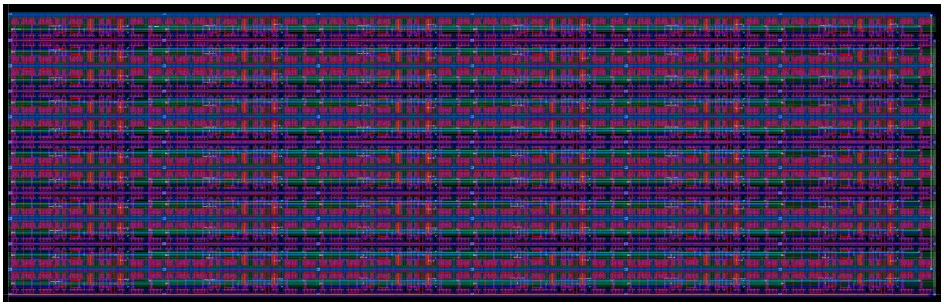

**Figure 1.30:** *Id +Ex Stage Layout*

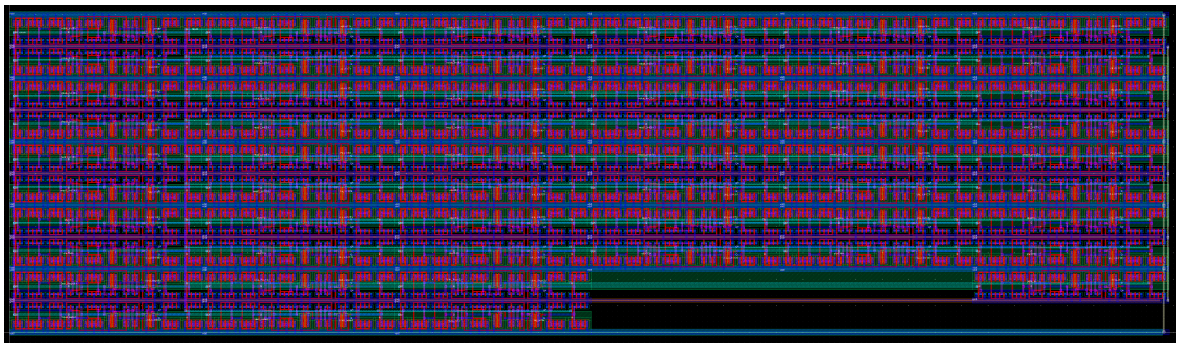

**Figure 1.31:** *Ex+Mem Stage Layout*

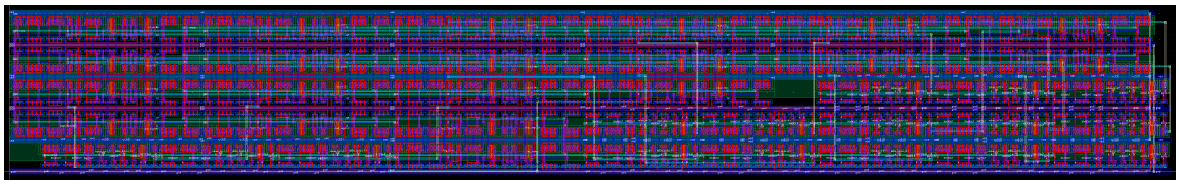

**Figure 1.32:** *Mem+WB stage layout*

### 1.4.3 LVS Result

```
1 @(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $
2
3 Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
4 Like matching is enabled.
5 Net swapping is enabled.
6 Using terminal names as correspondence points.
7 Compiling Diva LVS rules...
8
9     Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
```

```
      count
       862              nets
       134              terminals
       1980             pmos
       1980             nmos

   Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
      count
       862              nets
       136              terminals
       924              pmos
       924              nmos


   Terminal correspondence points
   N781       N53        ADD_en
   N853       N82        AND_en
   N826       N72        Data_in
   N759       N46        Dest_Reg<0>
   N745       N134       Dest_Reg<1>
   N733       N119       Dest_Reg<2>
   N809       N145       Dest_value<0>
   N791       N71        Dest_value<1>
   N776       N133       Dest_value<2>
   N836       N55        MUL_en
   N729       N78        Mem_select
   N808       N83        OR_en
   N751       N69        Reg1<0>
   N800       N59        Reg1<10>
   N782       N90        Reg1<11>
   N766       N89        Reg1<12>
   N752       N88        Reg1<13>
   N742       N87        Reg1<14>
   N728       N70        Reg1<15>
   N741       N68        Reg1<1>
   N860       N67        Reg1<2>
   N848       N66        Reg1<3>
   N831       N65        Reg1<4>
   N813       N64        Reg1<5>
   N793       N63        Reg1<6>
   N778       N62        Reg1<7>
   N761       N61        Reg1<8>
   N748       N60        Reg1<9>
   N833       N140       Reg2<0>
   N840       N95        Reg2<10>
   N823       N94        Reg2<11>
   N804       N93        Reg2<12>
   N786       N92        Reg2<13>
   N771       N91        Reg2<14>
   N758       N86        Reg2<15>
   N814       N144       Reg2<1>
   N797       N105       Reg2<2>
   N780       N104       Reg2<3>
   N765       N103       Reg2<4>
   N749       N102       Reg2<5>
   N738       N101       Reg2<6>
```

```
    N857        N99         Reg2<7>
    N845        N97         Reg2<8>
    N828        N96         Reg2<9>
    N842        N79         Reg_write
    N796        N54         SUB_en
    N815        N84         Shift_dir
    N736        N56         Shift_en
    N754        N81         VR_select
    N849        N77         Value_select
    N764        N136        add_en
    N837        N142        and_en
    N774        N117        clock
    N832        N6          data_in
    N834        N36         dest_reg<0>
    N817        N35         dest_reg<1>
    N798        N73         dest_reg<2>
    N795        N120        dest_value<0>
    N779        N146        dest_value<1>
    N763        N98         dest_value<2>
    N858        N43         mem_select
    N812        N137        mul_en
    N773        N143        or_en
    N777        N129        read1_in<0>
    N838        N21         read1_in<10>
    N821        N20         read1_in<11>
    N801        N17         read1_in<12>
    N783        N132        read1_in<13>
    N767        N131        read1_in<14>
    N753        N135        read1_in<15>
    N762        N128        read1_in<1>
    N747        N127        read1_in<2>
    N735        N126        read1_in<3>
    N855        N125        read1_in<4>
    N843        N100        read1_in<5>
    N825        N16         read1_in<6>
    N806        N15         read1_in<7>
    N788        N14         read1_in<8>
    N772        N22         read1_in<9>
    N760        N80         read2_in<0>
    N750        N25         read2_in<10>
    N739        N24         read2_in<11>
    N859        N23         read2_in<12>
    N846        N147        read2_in<13>
    N829        N148        read2_in<14>
    N810        N130        read2_in<15>
    N746        N34         read2_in<1>
    N734        N33         read2_in<2>
    N854        N32         read2_in<3>
    N839        N31         read2_in<4>
    N822        N30         read2_in<5>
    N803        N29         read2_in<6>
    N785        N28         read2_in<7>
    N770        N27         read2_in<8>
    N757        N26         read2_in<9>
    N816        N4          reg_write
    N851        N118        reset
```

```
    N789        N5          shift_dir
    N819        N138        shift_en
    N775        N149        sub_en
    N802        N52         value<0>
    N740        N42         value<10>
    N861        N41         value<11>
    N847        N40         value<12>
    N830        N39         value<13>
    N811        N38         value<14>
    N792        N37         value<15>
    N784        N51         value<1>
    N769        N50         value<2>
    N756        N49         value<3>
    N744        N48         value<4>
    N731        N47         value<5>
    N850        N76         value<6>
    N835        N75         value<7>
    N818        N74         value<8>
    N799        N58         value<9>
    N732        N123        value_in<0>
    N737        N19         value_in<10>
    N856        N18         value_in<11>
    N844        N12         value_in<12>
    N827        N10         value_in<13>
    N807        N9          value_in<14>
    N790        N2          value_in<15>
    N852        N122        value_in<1>
    N841        N13         value_in<2>
    N824        N11         value_in<3>
    N805        N7          value_in<4>
    N787        N116        value_in<5>
    N768        N115        value_in<6>
    N755        N114        value_in<7>
    N743        N113        value_in<8>
    N730        N112        value_in<9>
    N820        N85         value_select
    N794        N141        vr_select

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                                layout  schematic
                                  instances
        un-matched                0       0
        rewired                   0       0
        size errors               0       0
        pruned                    0       0
        active                    3960    1848
        total                     3960    1848

                                    nets
        un-matched                0       0
```

```
        merged                  0       0
        pruned                  0       0
        active                  862     862
        total                   862     862

                                terminals
        un-matched              0       0
        matched but
        different type          0       0
        total                   134     136


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

Code Listing 1.11: ID+EX Stage Register LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...
```

```
8
9     Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
10       count
11        719              nets
12        114              terminals
13        1650             pmos
14        1650             nmos
15
16    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
17       count
18        719              nets
19        114              terminals
20        770              pmos
21        770              nmos
22
23
24    Terminal correspondence points
25    N688       N61        Data_in
26    N634       N43        Dest_Reg<0>
27    N623       N42        Dest_Reg<1>
28    N610       N41        Dest_Reg<2>
29    N606       N63        Mem_select
30    N629       N80        Reg1<0>
31    N667       N70        Reg1<10>
32    N653       N69        Reg1<11>
33    N640       N68        Reg1<12>
34    N630       N67        Reg1<13>
35    N620       N66        Reg1<14>
36    N605       N65        Reg1<15>
37    N618       N79        Reg1<1>
38    N716       N78        Reg1<2>
39    N705       N77        Reg1<3>
40    N692       N76        Reg1<4>
41    N678       N75        Reg1<5>
42    N663       N74        Reg1<6>
43    N650       N73        Reg1<7>
44    N635       N72        Reg1<8>
45    N626       N71        Reg1<9>
46    N699       N64        Reg_write
47    N707       N62        Value_select
48    N676       N96        alu<0>
49    N713       N86        alu<10>
50    N703       N85        alu<11>
51    N690       N84        alu<12>
52    N674       N83        alu<13>
53    N659       N82        alu<14>
54    N645       N81        alu<15>
55    N661       N95        alu<1>
56    N649       N94        alu<2>
57    N638       N93        alu<3>
58    N625       N92        alu<4>
59    N612       N91        alu<5>
60    N712       N90        alu<6>
61    N701       N89        alu<7>
62    N687       N88        alu<8>
63    N672       N87        alu<9>
```

```
    N616        N105        alu_out<0>
    N679        N115        alu_out<10>
    N664        N116        alu_out<11>
    N652        N100        alu_out<12>
    N639        N99         alu_out<13>
    N628        N98         alu_out<14>
    N615        N97         alu_out<15>
    N717        N106        alu_out<1>
    N706        N107        alu_out<2>
    N694        N108        alu_out<3>
    N675        N109        alu_out<4>
    N660        N110        alu_out<5>
    N647        N111        alu_out<6>
    N636        N112        alu_out<7>
    N627        N113        alu_out<8>
    N613        N114        alu_out<9>
    N646        N21         clock
    N693        N104        data_in
    N695        N4          dest_reg<0>
    N681        N3          dest_reg<1>
    N665        N2          dest_reg<2>
    N619        N1          gnd!
    N715        N102        mem_select
    N648        N60         read1_in<0>
    N697        N50         read1_in<10>
    N684        N49         read1_in<11>
    N668        N48         read1_in<12>
    N654        N47         read1_in<13>
    N641        N46         read1_in<14>
    N631        N45         read1_in<15>
    N637        N59         read1_in<1>
    N624        N58         read1_in<2>
    N611        N57         read1_in<3>
    N711        N56         read1_in<4>
    N700        N55         read1_in<5>
    N686        N54         read1_in<6>
    N671        N53         read1_in<7>
    N657        N52         read1_in<8>
    N644        N51         read1_in<9>
    N680        N27         reg_write
    N709        N44         reset
    N669        N20         value<0>
    N617        N10         value<10>
    N718        N9          value<11>
    N704        N8          value<12>
    N691        N7          value<13>
    N677        N6          value<14>
    N662        N5          value<15>
    N655        N19         value<1>
    N643        N18         value<2>
    N633        N17         value<3>
    N622        N16         value<4>
    N608        N15         value<5>
    N708        N14         value<6>
    N696        N13         value<7>
    N682        N12         value<8>
```

```
    N666        N11         value<9>
    N609        N38         value_in<0>
    N614        N28         value_in<10>
    N714        N26         value_in<11>
    N702        N25         value_in<12>
    N689        N24         value_in<13>
    N673        N23         value_in<14>
    N658        N22         value_in<15>
    N710        N37         value_in<1>
    N698        N36         value_in<2>
    N685        N35         value_in<3>
    N670        N34         value_in<4>
    N656        N33         value_in<5>
    N642        N32         value_in<6>
    N632        N31         value_in<7>
    N621        N30         value_in<8>
    N607        N29         value_in<9>
    N683        N103        value_select
    N651        N0          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   3300    1540
        total                    3300    1540

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   719     719
        total                    719     719

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    114     114


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

```

```
mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

Code Listing 1.12: EX+MEM Stage Register LVS Result

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-11/anhv/EE477_VLSI/cds/LVS -l -s -t /home/scf-11/anhv/EE477_VLSI/
    cds/LVS/layout /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout/netlist
       count
         473             nets
         61              terminals
         904             pmos
         904             nmos

    Net-list summary for /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic/netlist
       count
         473             nets
         61              terminals
         472             pmos
         472             nmos


    Terminal correspondence points
    N428       N58        Dest_Reg<0>
```

```
    N424        N40         Dest_Reg<1>
    N416        N41         Dest_Reg<2>
    N465        N30         MEM_data<0>
    N417        N7          MEM_data<10>
    N469        N6          MEM_data<11>
    N460        N5          MEM_data<12>
    N453        N4          MEM_data<13>
    N445        N3          MEM_data<14>
    N437        N2          MEM_data<15>
    N456        N29         MEM_data<1>
    N448        N24         MEM_data<2>
    N440        N22         MEM_data<3>
    N433        N21         MEM_data<4>
    N429        N20         MEM_data<5>
    N425        N13         MEM_data<6>
    N419        N12         MEM_data<7>
    N470        N11         MEM_data<8>
    N462        N10         MEM_data<9>
    N413        N39         Mem_select
    N461        N32         Reg_write
    N432        N42         clock
    N457        N64         dest_reg<0>
    N450        N65         dest_reg<1>
    N441        N66         dest_reg<2>
    N421        N1          gnd!
    N449        N31         reg_write
    N467        N63         reset
    N444        N67         value<0>
    N420        N77         value<10>
    N472        N52         value<11>
    N464        N62         value<12>
    N455        N59         value<13>
    N447        N60         value<14>
    N439        N61         value<15>
    N436        N68         value<1>
    N431        N69         value<2>
    N427        N70         value<3>
    N423        N71         value<4>
    N414        N72         value<5>
    N466        N73         value<6>
    N458        N74         value<7>
    N451        N75         value<8>
    N442        N76         value<9>
    N415        N35         value_in<0>
    N418        N8          value_in<10>
    N471        N53         value_in<11>
    N463        N54         value_in<12>
    N454        N55         value_in<13>
    N446        N56         value_in<14>
    N438        N57         value_in<15>
    N468        N34         value_in<1>
    N459        N33         value_in<2>
    N452        N19         value_in<3>
    N443        N18         value_in<4>
    N435        N17         value_in<5>
    N430        N16         value_in<6>
```

```
    N426        N15         value_in<7>
    N422        N14         value_in<8>
    N412        N9          value_in<9>
    N434        N0          vdd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   1808    944
        total                    1808    944

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   0       0
        active                   473     473
        total                    473     473

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    61      61


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-11/anhv/EE477_VLSI/cds/LVS/layout

devbad.out:

```

```
netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

Code Listing 1.13: MEM+WB Stage Register LVS Result

## 1.5 CPU

### 1.5.1 Schematic Design

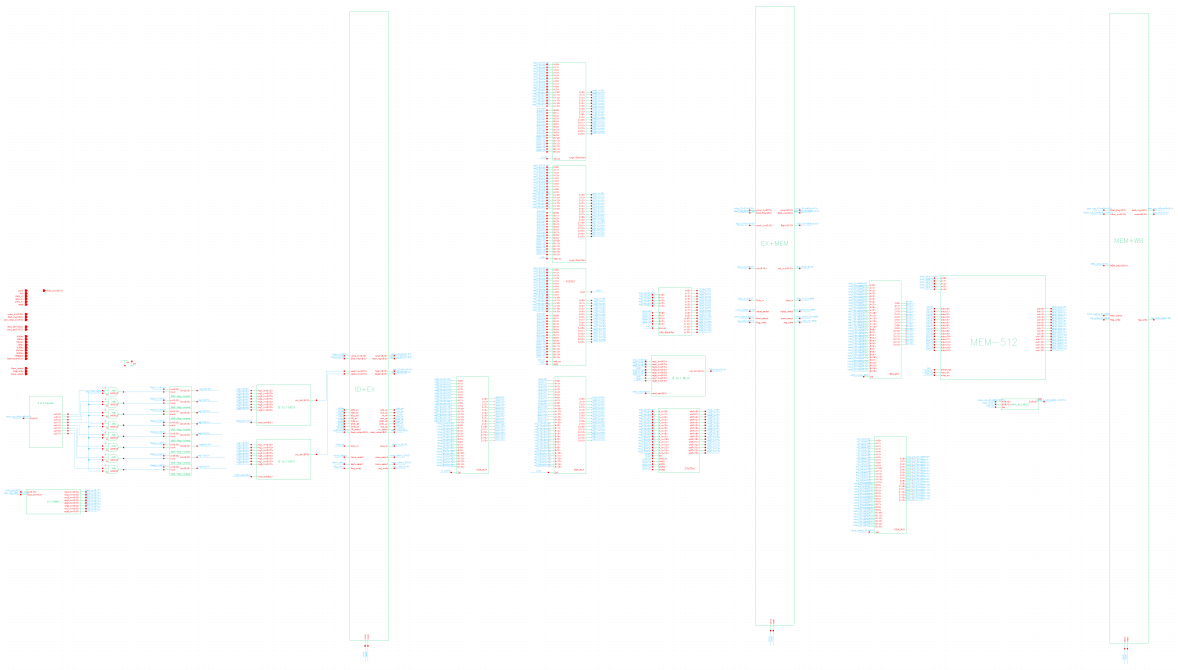


**Figure 1.33:** *CPU Schematic*

### 1.5.2 Layout Design

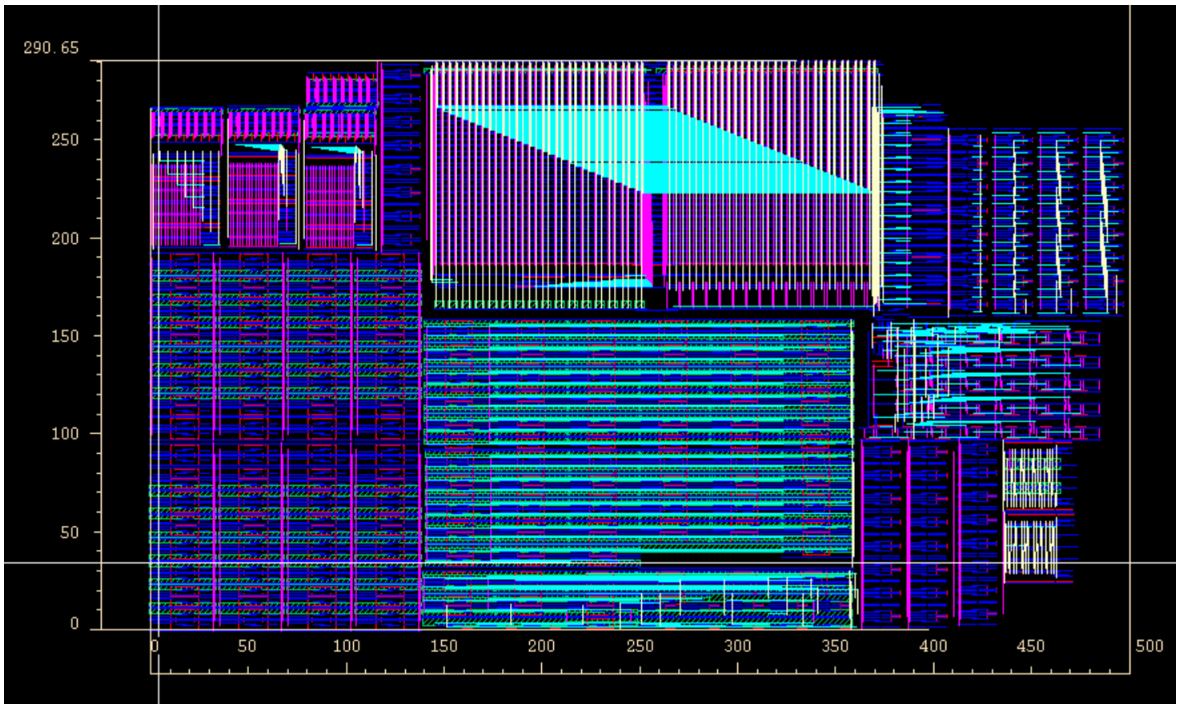


**Figure 1.34:** *CPU layout*

## 1.6 Python Code List

### 1.6.1 Front-End Part

For Python code optimization part, we applied the out-of-order algorithm, that is exchange the order of instructions to avoid the data hazard and keep the functionality of the program.

#### Instruction Decoding

The program can be divided into three parts. The first part is to decode the input instructions and give the instructions in normal order. Here shows the Python code.

```
def Five_bit(addr):
  if len(addr)<6:
    for i in range(5-len(addr)):
      addr = '0'+addr
    return addr

def Three_bit(addr):
  if len(addr)<4:
    for i in range(3-len(addr)):
      addr = '0'+addr
    return addr

def HexToBin(input_addr):
  if input_addr[-1] == 'B':
    return input_addr[0:-1]
```

```
16    else:
17      addr =  bin(int(input_addr[0:-1], 16))[2:]
18      return Five_bit(addr)
19
20  def DecToBin(input_addr):
21    return Three_bit(bin(int(input_addr))[2:])
22
23  def STOREI(splited):
24    a=[]
25    if splited[1][-1] == '}':
26      burst = splited[1][-2]
27      address = HexToBin(splited[2])
28    else:
29      burst = '1'
30      address = HexToBin(splited[1])
31    if (burst == '1' or burst == '2' or burst == '4') != True:
32      return 'Error1'
33    else:
34      if  (burst == '1' or (address[-1] == '0' and burst == '2') or (address[-2:]
        == '00' and burst == '4')) != True:
35        # print burst
36        # print address[-1]
37        return 'Error2'
38      else:
39        if burst == '1':
40          a.append(['STOREI','External',splited[-1][1:],'','','word',address])
41          return a
42        if burst == '2':
43          a.append(['STOREI','External',splited[-2][1:],'','','word',address])
44          a.append(['STOREI','External',splited[-1][2:-1],'','','word',Five_bit(
      str(int(address)+1))])
45          return a
46        if burst == '4':
47          a.append(['STOREI','External',splited[-4][1:],'','','word',address])
48          a.append(['STOREI','External',splited[-3][2:-1],'','','word',Five_bit(
      str(int(address)+1))])
49          a.append(['STOREI','External',splited[-2][2:-1],'','','word',Five_bit(
      str(int(address)+10))])
50          a.append(['STOREI','External',splited[-1][2:-1],'','','word',Five_bit(
      str(int(address)+11))])
51          return a
52
53  def STORE(splited):
54    return ['STORE','Reg',(splited[2][1]),'','','word',HexToBin(splited[1])]
55
56  def LOADI(splited):
57    return ['LOADI','External',splited[-1][1:],'','','Reg',(splited[1][1])]
58
59  def LOAD(splited):
60    return ['LOAD','word',HexToBin(splited[-1]),'','','Reg',(splited[1][1])]
61
62  def AND(splited):
63    return ['AND','Reg',(splited[2][1]),'Reg',(splited[3][1]),'Reg',(splited
      [1][1])]
64
65  def ANDI(splited):
```

```
66    return ['ANDI','Reg',(splited[2][1]),'External',splited[-1][0:],'Reg',(
        splited[1][1])]
67
68  def OR(splited):
69    return ['OR','Reg',(splited[2][1]),'Reg',(splited[3][1]),'Reg',(splited
        [1][1])]
70
71  def ORI(splited):
72    return ['ORI','Reg',(splited[2][1]),'External',splited[-1][0:],'Reg',(splited
        [1][1])]
73
74  def ADD(splited):
75    return ['ADD','Reg',(splited[2][1]),'Reg',(splited[3][1]),'Reg',(splited
        [1][1])]
76
77  def ADDI(splited):
78    return ['ADDI','Reg',(splited[2][1]),'External',splited[-1][0:],'Reg',(
        splited[1][1])]
79
80  def NOP(splited):
81    return ['NOP','','','','','','']
82
83  def MUL(splited):
84    return['MUL','Reg',(splited[2][1]),'Reg',(splited[-1][1:]),'Reg',(splited
        [1][1])]
85
86  def MULI(splited):
87    return ['MULI','Reg',(splited[2][1]),'External',splited[-1][1:],'Reg',(
        splited[1][1])]
88
89  def MIN(splited):
90    return['MIN','Reg',(splited[2][1]),'Reg',(splited[-1][1:]),'Reg',(splited
        [1][1])]
91
92  def MINI(splited):
93    return ['MINI','Reg',(splited[2][1]),'External',splited[-1][1:],'Reg',(
        splited[1][1])]
94
95  def SFL(splited):
96    return ['SFL','Reg',(splited[2][1]),'External',splited[-1][1:],'Reg',(splited
        [1][1])]
97
98  def SFR(splited):
99    return ['SFR','Reg',(splited[2][1]),'External',splited[-1][1:],'Reg',(splited
        [1][1])]
100
101 def Bubble(input_op1,input_op2):
102   bubble = False
103   if (input_op1[5] == input_op2[1]) and (input_op1[5] != ''):
104     if input_op1[6] == input_op2[2]:
105       bubble = True
106   if input_op1[5] == input_op2[3] and (input_op1[5] != ''):
107     if input_op1[6] == input_op2[4]:
108       bubble = True
109   return bubble
110
```

```
111 def write_array(files,array):
112   for i in array:
113     files.write(i+'\t')
114   files.write('\n')
115 
116 def write_NOP(files,bubble):
117   if bubble == True:
118     write_array(files,['NOP','','','','','','',''])
119     write_array(files,['NOP','','','','','','',''])
120     write_array(files,['NOP','','','','','','',''])
121 
122 FileIn = open('input.txt','r')
123 FileOut = open('output.txt','w')
124 
125 ## Initialization
126 last = ['','','','','','','','']
127 ## Main Loop
128 for words in FileIn:
129   bubble = False
130   input_op = words.split()
131   if input_op[0] == 'NOP':
132     new = NOP(input_op)
133     last = new
134     write_array(FileOut,last)
135   elif input_op[0]=='STOREI':
136     if STOREI(input_op) == 'Error1':
137       FileOut.write('Error1'+'\n')
138     elif STOREI(input_op) == 'Error2':
139       FileOut.write('Error2'+'\n')
140     else:
141       new = STOREI(input_op)[0]
142       last = STOREI(input_op)[-1]
143       for i in STOREI(input_op):
144         write_array(FileOut,i)
145   elif input_op[0]=='STORE':
146     new = STORE(input_op)
147     last = new
148     write_array(FileOut,last)
149   elif input_op[0]=='LOADI':
150     new = LOADI(input_op)
151     last = new
152     write_array(FileOut,last)
153   elif input_op[0]=='LOAD':
154     new = LOAD(input_op)
155     last = new
156     write_array(FileOut,last)
157   elif input_op[0]=='AND':
158     new = AND(input_op)
159     last = new
160     write_array(FileOut,last)
161   elif input_op[0]=='ANDI':
162     new = ANDI(input_op)
163     last = new
164     write_array(FileOut,last)
165   elif input_op[0]=='OR':
166     new = OR(input_op)
```

```
167     last = new
168     write_array(FileOut,last)
169   elif input_op[0]=='ORI':
170     new = ORI(input_op)
171     last = new
172     write_array(FileOut,last)
173   elif input_op[0]=='ADD':
174     new = ADD(input_op)
175     last = new
176     write_array(FileOut,last)
177   elif input_op[0]=='ADDI':
178     new = ADDI(input_op)
179     last = new
180     write_array(FileOut,last)
181   elif input_op[0]=='MUL':
182     new = MUL(input_op)
183     last = new
184     write_array(FileOut,last)
185   elif input_op[0]=='MULI':
186     new = MULI(input_op)
187     last = new
188     write_array(FileOut,last)
189   elif input_op[0]=='MIN':
190     new = MIN(input_op)
191     last = new
192     write_array(FileOut,last)
193   elif input_op[0]=='MINI':
194     new = MINI(input_op)
195     last = new
196     write_array(FileOut,last)
197   elif input_op[0]=='SFL':
198     new = SFL(input_op)
199     last = new
200     write_array(FileOut,last)
201   elif input_op[0]=='SFR':
202     new = SFR(input_op)
203     last = new
204     write_array(FileOut,last)
205
206 FileIn.close()
207 FileOut.close()
```

Code Listing 1.14: Part 1: Instruction Decoding

### Out-of-Order Algorithm

Here we applied the out-of-order concept, that is swapping the positions of two instructions while keeping the functionality of the pro-gram,then we can avoid insert the "NOP" instruction and save some clocks.And if there is no way to solve the data dependency with swapping the instructions, it will insert minimal number of "NOP" instructions. Here gives the Python code.

```
1 def write_array(files,array):
2   for i in array:
3     files.write(i+'\t')
```

```
  files.write('\n')

def conflict(input_op1,input_op2):
  bubble = False
  if (input_op1[5] == input_op2[1]) and (input_op1[5] != ''):
    if input_op1[6] == input_op2[2]:
      bubble = True
  if input_op1[5] == input_op2[3] and (input_op1[5] != ''):
    if input_op1[6] == input_op2[4]:
      bubble = True
  return bubble

def findit(instr,array):
  find = False
  for i in array:
    if conflict(instr.split('\t'),i.split('\t')) == False:
      find = True
      for j in array:
        if j == i:
          break
        elif conflict(j.split('\t'),i.split('\t')) == True:
          find = False
          break
      if find == True:
        break
  if find == False:
    return None
  elif find == True:
    return i

## swap position of a and b in list
def swaplist(lists,a,b):
  counta = countb = 0
  finda = False
  findb = False
  for i in lists:
    if i == a:
      finda == True
      break
    if finda == False:
      counta = counta + 1
  for i in lists:
    if i == b:
      findb == True
      break
    if findb == False:
      countb = countb +1
  lists[counta] = b
  lists[countb] = a
  return lists

FileIn = open('output.txt','r')
FileOut = open('output_OFO.txt','w')
A=[]
B=[]
for i in FileIn:
```

```
60    A.append(i)
61
62  while A != []:
63    if len(A) == 3:
64      tommy = A[0]
65      B.append(tommy)
66      A = A[1:]
67      for count in range(0,2):
68        if findit(tommy,A[count:]) != None:
69          A = swaplist(A,A[count],findit(tommy,A[count:]))
70        else:
71          for count_NOP in range(3-count):
72            A.insert(count,'NOP'+'\t'+''+'\t'+''+'\t'+''+'\t'+''+'\t'+''+'\t'+''+
      '\n')
73          break
74    elif len(A) == 2:
75      tommy = A[0]
76      B.append(tommy)
77      A = A[1:]
78      for count in range(0,1):
79        if findit(tommy,A[count:]) != None:
80          A = swaplist(A,A[count],findit(tommy,A[count:]))
81        else:
82          for count_NOP in range(3-count):
83            A.insert(count,'NOP'+'\t'+''+'\t'+''+'\t'+''+'\t'+''+'\t'+''+'\t'+''+
      '\n')
84          break
85    elif len(A) == 1:
86      tommy = A[0]
87      B.append(tommy)
88      A = A[1:]
89    else:
90      tommy = A[0]
91      B.append(tommy)
92      A = A[1:]
93      for count in range(0,3):
94        if findit(tommy,A[count:]) != None:
95          A = swaplist(A,A[count],findit(tommy,A[count:]))
96        else:
97          for count_NOP in range(3-count):
98            A.insert(count,'NOP'+'\t'+''+'\t'+''+'\t'+''+'\t'+''+'\t'+''+'\t'+''+
      '\n')
99          break
100
101 for i in B:
102   FileOut.write(i)
```

Code Listing 1.15: Part 2: Out-of-Order Algorithm Implementation

### Vector File Generation

Part 3 is for compiling the output file generated by Part 2 to readable vector file. Compared with in Phase 1, we divided the time unit to 10, say if the clock is 10ns, then the time unit in the vector file will be 1ns, this will provide more maneuverability of the signal, which can help saving power and clock period.

For example, when doing STORE and LOAD operations, we will need to operate the "precharge" signal and "read_en" or "write_en" signals. In Phase 1, we kept the "precharge" signal the same with the clock signal, which means in every cycle, the Memory's bitline and bitline_bar would be charged and discharged, which will waste a lot of switching power and static power. And for now, since we can manipulate the "precharge" signal more precisely, we keep it as "1" until there are Memory-related instructions are given, this helps save a lot of power.

The new vector file generation Python code is shown below.

```
def write_array(files,array):
  for i in array:
    files.write(i+'\t')
  files.write('\n')

def Four_bit(addr):
  if len(addr)<5:
    for i in range(4-len(addr)):
      addr = '0'+addr
    return addr

def heappop(array):
  array = array[1:]
  return array[0]

FileIn = open('output_OFO.txt','r')
FileOut2 = open('control.vec','w')
FileOut2.write('radix'+'\t'+'4'+'\t'+'4'+'\t'+'4'+'\t'+'4'+'\t'+'1'+'\t'+'1'+'\
    t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'3'+'\t'+'3'+'\t'+'3'+'\t'+'1'+'\t'+'1'+'\t'+
    '1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'
    +'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\t'+'1'+'\n')
FileOut2.write('io'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+
    'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'
    +'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\
    t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\t'+'i'+'\n')
FileOut2.write('vname'+'\t'+'value_in<[15:12]>'+'\t'+'value_in<[11:8]>'+'\t'+'
    value_in<[7:4]>'+'\t'+'value_in<[3:0]>'+'\t'+'instr_addr_in<4>'+'\t'+'
    instr_addr_in<3>'+'\t'+'instr_addr_in<2>'+'\t'+'instr_addr_in<1>'+'\t'+'
    instr_addr_in<0>'+'\t'+'Dest_reg<[2:0]>'+'\t'+'read_sel1<[2:0]>'+'\t'+'
    read_sel2<[2:0]>'+'\t'+'VRselect'+'\t'+'ANDen'+'\t'+'OREn'+'\t'+'ADDen'+'\t
    '+'SUBen'+'\t'+'MULen'+'\t'+'Shiften'+'\t'+'Shiftdir'+'\t'+'Destvalue<2>'+'
    \t'+'Destvalue<1>'+'\t'+'Destvalue<0>'+'\t'+'Data_in'+'\t'+'write_en'+'\t'+
    'read_en'+'\t'+'pre'+'\t'+'Value_select'+'\t'+'Mem_select'+'\t'+'Reg_write'
    +'\n')
FileOut2.write('vih'+'\t'+'1.8'+'\n')
FileOut2.write('slope'+'\t'+'0.001'+'\n')
FileOut2.write('tunit'+'\t'+'0.5ns'+'\n')
Value = '0000'
Address = '00000'
Dest_Reg = Reg1 = Reg2 = '0'
Dest_value = '000'
VR_select = AND_en = OR_en = ADD_en = SUB = MUL_en = Data_in = Write_en =
    Read_en = Precharge = Value_select = MEM_seclect = Reg_write = '0'
SHIFT_en = '0'
SHIFT_dir = '0'
command=[Value[0], Value[1], Value[2], Value[3], Address[4], Address[3],
```

```
    Address[2], Address[1], Address[0], Dest_Reg, Reg1, Reg2, VR_select, AND_en
    , OR_en, ADD_en, SUB, MUL_en, SHIFT_en, SHIFT_dir, Dest_value[0],
    Dest_value[1], Dest_value[2], Data_in, Write_en, Read_en, Precharge,
    Value_select, MEM_seclect, Reg_write]
Write_en_heap = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']
Read_en_heap = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']
Precharge_heap = ['1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1','1']
Addr_0_heap = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']
Addr_1_heap = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']
Addr_2_heap = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']
Addr_3_heap = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']
Addr_4_heap = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']
count = 0
for i in FileIn:
  last = i.split()
  if last[0] == 'NOP':
    for i in range(10):
      count = count + 1
      #print 'NOP'
      command[13] = command[14] = command[15] = command[16] = command[17] =
    command[18] = command[19] = command[-1] = '0'
      ## Get value from heap
      #print Write_en_heap
      command[24] = Write_en_heap[0]
      Write_en_heap = Write_en_heap[1:]
      command[25] = Read_en_heap[0]
      Read_en_heap = Read_en_heap[1:]
      command[26] = Precharge_heap[0]
      Precharge_heap = Precharge_heap[1:]
      command[4] = Addr_0_heap[0]
      Addr_0_heap = Addr_0_heap[1:]
      command[5] = Addr_1_heap[0]
      Addr_1_heap = Addr_1_heap[1:]
      command[6] = Addr_2_heap[0]
      Addr_2_heap = Addr_2_heap[1:]
      command[7] = Addr_3_heap[0]
      Addr_3_heap = Addr_3_heap[1:]
      command[8] = Addr_4_heap[0]
      Addr_4_heap = Addr_4_heap[1:]
      ## Insert value to heap
      Write_en_heap.append('0')
      Read_en_heap.append('0')
      Precharge_heap.append('1')
      Addr_0_heap.append('0')
      Addr_1_heap.append('0')
      Addr_2_heap.append('0')
      Addr_3_heap.append('0')
      Addr_4_heap.append('0')
```

```
75        write_array(FileOut2,[str(count)]+command)
76    elif last[0] == 'STOREI':
77      for i in range(10):
78        count = count + 1
79        command[0] = last[2][0]
80        command[1] = last[2][1]
81        command[2] = last[2][2]
82        command[3] = last[2][3]
83        command[13] = command[14] = command[15] = command[16] = command[17] =
      command[18] = command[19] = command[29] = '0'
84        command[23] = '1'
85        ## Get value from heap
86        command[24] = Write_en_heap[0]
87        Write_en_heap = Write_en_heap[1:]
88        command[25] = Read_en_heap[0]
89        Read_en_heap = Read_en_heap[1:]
90        command[26] = Precharge_heap[0]
91        Precharge_heap = Precharge_heap[1:]
92        command[4] = Addr_0_heap[0]
93        Addr_0_heap = Addr_0_heap[1:]
94        command[5] = Addr_1_heap[0]
95        Addr_1_heap = Addr_1_heap[1:]
96        command[6] = Addr_2_heap[0]
97        Addr_2_heap = Addr_2_heap[1:]
98        command[7] = Addr_3_heap[0]
99        Addr_3_heap = Addr_3_heap[1:]
100       command[8] = Addr_4_heap[0]
101       Addr_4_heap = Addr_4_heap[1:]
102       if (i == 0) == True:
103         Write_en_heap.append('0')
104         Read_en_heap.append('0')
105         Precharge_heap.append('0')
106         Addr_0_heap.append('0')
107         Addr_1_heap.append('0')
108         Addr_2_heap.append('0')
109         Addr_3_heap.append('0')
110         Addr_4_heap.append('0')
111       elif (i == 1) == True:
112         Write_en_heap.append('0')
113         Read_en_heap.append('0')
114         Precharge_heap.append('0')
115         Addr_0_heap.append(last[-1][0])
116         Addr_1_heap.append(last[-1][1])
117         Addr_2_heap.append(last[-1][2])
118         Addr_3_heap.append(last[-1][3])
119         Addr_4_heap.append(last[-1][4])
120       elif (i >= 2 and i <=4) == True:
121         Write_en_heap.append('0')
122         Read_en_heap.append('0')
123         Precharge_heap.append('1')
124         Addr_0_heap.append(last[-1][0])
125         Addr_1_heap.append(last[-1][1])
126         Addr_2_heap.append(last[-1][2])
127         Addr_3_heap.append(last[-1][3])
128         Addr_4_heap.append(last[-1][4])
129       elif (i >=5 and i <=8) == True:
```

```
130         ## Insert value to heap
131         Write_en_heap.append('1')
132         Read_en_heap.append('0')
133         Precharge_heap.append('1')
134         Addr_0_heap.append(last[-1][0])
135         Addr_1_heap.append(last[-1][1])
136         Addr_2_heap.append(last[-1][2])
137         Addr_3_heap.append(last[-1][3])
138         Addr_4_heap.append(last[-1][4])
139       else:
140         Write_en_heap.append('0')
141         Read_en_heap.append('0')
142         Precharge_heap.append('1')
143         Addr_0_heap.append('0')
144         Addr_1_heap.append('0')
145         Addr_2_heap.append('0')
146         Addr_3_heap.append('0')
147         Addr_4_heap.append('0')
148       write_array(FileOut2,[str(count)]+command)
149   elif last[0] == 'STORE':
150     for i in range(10):
151       count = count + 1
152       command[10] = last[2]
153       command[13] = command[14] = command[15] = command[16] = command[17] =
      command[18] = command[19] = command[23] = command[29] = '0'
154       ## Get value from heap
155       command[24] = Write_en_heap[0]
156       Write_en_heap = Write_en_heap[1:]
157       command[25] = Read_en_heap[0]
158       Read_en_heap = Read_en_heap[1:]
159       command[26] = Precharge_heap[0]
160       Precharge_heap = Precharge_heap[1:]
161       command[4] = Addr_0_heap[0]
162       Addr_0_heap = Addr_0_heap[1:]
163       command[5] = Addr_1_heap[0]
164       Addr_1_heap = Addr_1_heap[1:]
165       command[6] = Addr_2_heap[0]
166       Addr_2_heap = Addr_2_heap[1:]
167       command[7] = Addr_3_heap[0]
168       Addr_3_heap = Addr_3_heap[1:]
169       command[8] = Addr_4_heap[0]
170       Addr_4_heap = Addr_4_heap[1:]
171       if (i == 0) == True:
172         Write_en_heap.append('0')
173         Read_en_heap.append('0')
174         Precharge_heap.append('0')
175         Addr_0_heap.append('0')
176         Addr_1_heap.append('0')
177         Addr_2_heap.append('0')
178         Addr_3_heap.append('0')
179         Addr_4_heap.append('0')
180       elif (i == 1) == True:
181         Write_en_heap.append('0')
182         Read_en_heap.append('0')
183         Precharge_heap.append('0')
184         Addr_0_heap.append(last[-1][0])
```

```
185         Addr_1_heap.append(last[-1][1])
186         Addr_2_heap.append(last[-1][2])
187         Addr_3_heap.append(last[-1][3])
188         Addr_4_heap.append(last[-1][4])
189       elif (i >= 2 and i <=4) == True:
190         Write_en_heap.append('0')
191         Read_en_heap.append('0')
192         Precharge_heap.append('1')
193         Addr_0_heap.append(last[-1][0])
194         Addr_1_heap.append(last[-1][1])
195         Addr_2_heap.append(last[-1][2])
196         Addr_3_heap.append(last[-1][3])
197         Addr_4_heap.append(last[-1][4])
198       elif (i >=5 and i <=8) == True:
199         ## Insert value to heap
200         Write_en_heap.append('1')
201         Read_en_heap.append('0')
202         Precharge_heap.append('1')
203         Addr_0_heap.append(last[-1][0])
204         Addr_1_heap.append(last[-1][1])
205         Addr_2_heap.append(last[-1][2])
206         Addr_3_heap.append(last[-1][3])
207         Addr_4_heap.append(last[-1][4])
208       else:
209         Write_en_heap.append('0')
210         Read_en_heap.append('0')
211         Precharge_heap.append('1')
212         Addr_0_heap.append('0')
213         Addr_1_heap.append('0')
214         Addr_2_heap.append('0')
215         Addr_3_heap.append('0')
216         Addr_4_heap.append('0')
217       write_array(FileOut2,[str(count)]+command)
218   elif last[0] == 'LOADI':
219     for i in range(10):
220       count = count + 1
221       command[0] = last[2][0]
222       command[1] = last[2][1]
223       command[2] = last[2][2]
224       command[3] = last[2][3]
225       command[9] = last[-1]
226       command[13] = command[14] = command[15] = command[16] = command[17] =
    command[18] = command[19] = command[28] = '0'
227       command[27] = command[29] = '1'
228       ## Get value from heap
229       command[24] = Write_en_heap[0]
230       Write_en_heap = Write_en_heap[1:]
231       command[25] = Read_en_heap[0]
232       Read_en_heap = Read_en_heap[1:]
233       command[26] = Precharge_heap[0]
234       Precharge_heap = Precharge_heap[1:]
235       command[4] = Addr_0_heap[0]
236       Addr_0_heap = Addr_0_heap[1:]
237       command[5] = Addr_1_heap[0]
238       Addr_1_heap = Addr_1_heap[1:]
239       command[6] = Addr_2_heap[0]
```

```
240         Addr_2_heap = Addr_2_heap[1:]
241         command[7] = Addr_3_heap[0]
242         Addr_3_heap = Addr_3_heap[1:]
243         command[8] = Addr_4_heap[0]
244         Addr_4_heap = Addr_4_heap[1:]
245         ## Insert value to heap
246         Write_en_heap.append('0')
247         Read_en_heap.append('0')
248         Precharge_heap.append('1')
249         Addr_0_heap.append('0')
250         Addr_1_heap.append('0')
251         Addr_2_heap.append('0')
252         Addr_3_heap.append('0')
253         Addr_4_heap.append('0')
254         write_array(FileOut2,[str(count)]+command)
255     elif last[0] == 'LOAD':
256       for i in range(10):
257         count = count + 1
258         command[9] = last[-1]
259         command[13] = command[14] = command[15] = command[16] = command[17] =
      command[18] = command[19] = '0'
260         command[28] = command[29] = '1'
261         ## Get value from heap
262         command[24] = Write_en_heap[0]
263         Write_en_heap = Write_en_heap[1:]
264         command[25] = Read_en_heap[0]
265         Read_en_heap = Read_en_heap[1:]
266         command[26] = Precharge_heap[0]
267         Precharge_heap = Precharge_heap[1:]
268         command[4] = Addr_0_heap[0]
269         Addr_0_heap = Addr_0_heap[1:]
270         command[5] = Addr_1_heap[0]
271         Addr_1_heap = Addr_1_heap[1:]
272         command[6] = Addr_2_heap[0]
273         Addr_2_heap = Addr_2_heap[1:]
274         command[7] = Addr_3_heap[0]
275         Addr_3_heap = Addr_3_heap[1:]
276         command[8] = Addr_4_heap[0]
277         Addr_4_heap = Addr_4_heap[1:]
278         if (i == 0) == True:
279           Write_en_heap.append('0')
280           Read_en_heap.append('0')
281           Precharge_heap.append('0')
282           Addr_0_heap.append('0')
283           Addr_1_heap.append('0')
284           Addr_2_heap.append('0')
285           Addr_3_heap.append('0')
286           Addr_4_heap.append('0')
287         elif (i == 1) == True:
288           Write_en_heap.append('0')
289           Read_en_heap.append('0')
290           Precharge_heap.append('0')
291           Addr_0_heap.append(last[2][0])
292           Addr_1_heap.append(last[2][1])
293           Addr_2_heap.append(last[2][2])
294           Addr_3_heap.append(last[2][3])
```

```
295         Addr_4_heap.append(last[2][4])
296       elif (i >= 2 and i <=4) == True:
297         Write_en_heap.append('0')
298         Read_en_heap.append('0')
299         Precharge_heap.append('1')
300         Addr_0_heap.append(last[2][0])
301         Addr_1_heap.append(last[2][1])
302         Addr_2_heap.append(last[2][2])
303         Addr_3_heap.append(last[2][3])
304         Addr_4_heap.append(last[2][4])
305       elif (i >=5 and i <=8) == True:
306         ## Insert value to heap
307         Write_en_heap.append('0')
308         Read_en_heap.append('1')
309         Precharge_heap.append('1')
310         Addr_0_heap.append(last[2][0])
311         Addr_1_heap.append(last[2][1])
312         Addr_2_heap.append(last[2][2])
313         Addr_3_heap.append(last[2][3])
314         Addr_4_heap.append(last[2][4])
315       else:
316         Write_en_heap.append('0')
317         Read_en_heap.append('0')
318         Precharge_heap.append('1')
319         Addr_0_heap.append('0')
320         Addr_1_heap.append('0')
321         Addr_2_heap.append('0')
322         Addr_3_heap.append('0')
323         Addr_4_heap.append('0')
324       write_array(FileOut2,[str(count)]+command)
325   elif last[0] == 'AND':
326     for i in range(10):
327       count = count + 1
328       command[9] = last[6]
329       command[10] = last[2]
330       command[11] = last[4]
331       command[12] = command[14] = command[15] = command[16] = command[17] =
      command[18] = command[19] = command[27] = command[28] ='0'
332       command[13] = command[29] = '1'
333       command[20] = command[21] = command[22] = '0'
334       ## Get value from heap
335       command[24] = Write_en_heap[0]
336       Write_en_heap = Write_en_heap[1:]
337       command[25] = Read_en_heap[0]
338       Read_en_heap = Read_en_heap[1:]
339       command[26] = Precharge_heap[0]
340       Precharge_heap = Precharge_heap[1:]
341       command[4] = Addr_0_heap[0]
342       Addr_0_heap = Addr_0_heap[1:]
343       command[5] = Addr_1_heap[0]
344       Addr_1_heap = Addr_1_heap[1:]
345       command[6] = Addr_2_heap[0]
346       Addr_2_heap = Addr_2_heap[1:]
347       command[7] = Addr_3_heap[0]
348       Addr_3_heap = Addr_3_heap[1:]
349       command[8] = Addr_4_heap[0]
```

```
350         Addr_4_heap = Addr_4_heap[1:]
351         ## Insert value to heap
352         Write_en_heap.append('0')
353         Read_en_heap.append('0')
354         Precharge_heap.append('1')
355         Addr_0_heap.append('0')
356         Addr_1_heap.append('0')
357         Addr_2_heap.append('0')
358         Addr_3_heap.append('0')
359         Addr_4_heap.append('0')
360         write_array(FileOut2,[str(count)]+command)
361     elif last[0] == 'ANDI':
362       for i in range(10):
363         count = count + 1
364         command[0] = last[4][0]
365         command[1] = last[4][1]
366         command[2] = last[4][2]
367         command[3] = last[4][3]
368         command[9] = last[6]
369         command[10] = last[2]
370         command[14] = command[15] = command[16] = command[17] = command[18] =
      command[19] = command[27] = command[28] ='0'
371         command[12] = command[13] = command[29] = '1'
372         command[20] = command[21] = command[22] = '0'
373         ## Get value from heap
374         command[24] = Write_en_heap[0]
375         Write_en_heap = Write_en_heap[1:]
376         command[25] = Read_en_heap[0]
377         Read_en_heap = Read_en_heap[1:]
378         command[26] = Precharge_heap[0]
379         Precharge_heap = Precharge_heap[1:]
380         command[4] = Addr_0_heap[0]
381         Addr_0_heap = Addr_0_heap[1:]
382         command[5] = Addr_1_heap[0]
383         Addr_1_heap = Addr_1_heap[1:]
384         command[6] = Addr_2_heap[0]
385         Addr_2_heap = Addr_2_heap[1:]
386         command[7] = Addr_3_heap[0]
387         Addr_3_heap = Addr_3_heap[1:]
388         command[8] = Addr_4_heap[0]
389         Addr_4_heap = Addr_4_heap[1:]
390         ## Insert value to heap
391         Write_en_heap.append('0')
392         Read_en_heap.append('0')
393         Precharge_heap.append('1')
394         Addr_0_heap.append('0')
395         Addr_1_heap.append('0')
396         Addr_2_heap.append('0')
397         Addr_3_heap.append('0')
398         Addr_4_heap.append('0')
399         write_array(FileOut2,[str(count)]+command)
400     elif last[0] == 'OR':
401       for i in range(10):
402         count = count + 1
403         command[9] = last[6]
404         command[10] = last[2]
```

```
405         command[11] = last[4]
406         command[12] = command[13] = command[15] = command[16] = command[17] =
        command[18] = command[19] = command[27] = command[28] ='0'
407         command[14] = command[29] = '1'
408         command[20] = command[22] = '1'
409         command[21] = '0'
410         ## Get value from heap
411         command[24] = Write_en_heap[0]
412         Write_en_heap = Write_en_heap[1:]
413         command[25] = Read_en_heap[0]
414         Read_en_heap = Read_en_heap[1:]
415         command[26] = Precharge_heap[0]
416         Precharge_heap = Precharge_heap[1:]
417         command[4] = Addr_0_heap[0]
418         Addr_0_heap = Addr_0_heap[1:]
419         command[5] = Addr_1_heap[0]
420         Addr_1_heap = Addr_1_heap[1:]
421         command[6] = Addr_2_heap[0]
422         Addr_2_heap = Addr_2_heap[1:]
423         command[7] = Addr_3_heap[0]
424         Addr_3_heap = Addr_3_heap[1:]
425         command[8] = Addr_4_heap[0]
426         Addr_4_heap = Addr_4_heap[1:]
427         ## Insert value to heap
428         Write_en_heap.append('0')
429         Read_en_heap.append('0')
430         Precharge_heap.append('1')
431         Addr_0_heap.append('0')
432         Addr_1_heap.append('0')
433         Addr_2_heap.append('0')
434         Addr_3_heap.append('0')
435         Addr_4_heap.append('0')
436         write_array(FileOut2,[str(count)]+command)
437     elif last[0] == 'ORI':
438       for i in range(10):
439         count = count + 1
440         command[0] = last[4][0]
441         command[1] = last[4][1]
442         command[2] = last[4][2]
443         command[3] = last[4][3]
444         command[9] = last[6]
445         command[10] = last[2]
446         command[13] = command[15] = command[16] = command[17] = command[18] =
        command[19] = command[27] = command[28] ='0'
447         command[12] = command[14] = command[29] = '1'
448         command[20] = command[22] = '1'
449         command[21] = '0'
450         ## Get value from heap
451         command[24] = Write_en_heap[0]
452         Write_en_heap = Write_en_heap[1:]
453         command[25] = Read_en_heap[0]
454         Read_en_heap = Read_en_heap[1:]
455         command[26] = Precharge_heap[0]
456         Precharge_heap = Precharge_heap[1:]
457         command[4] = Addr_0_heap[0]
458         Addr_0_heap = Addr_0_heap[1:]
```

```
459         command[5] = Addr_1_heap[0]
460         Addr_1_heap = Addr_1_heap[1:]
461         command[6] = Addr_2_heap[0]
462         Addr_2_heap = Addr_2_heap[1:]
463         command[7] = Addr_3_heap[0]
464         Addr_3_heap = Addr_3_heap[1:]
465         command[8] = Addr_4_heap[0]
466         Addr_4_heap = Addr_4_heap[1:]
467         ## Insert value to heap
468         Write_en_heap.append('0')
469         Read_en_heap.append('0')
470         Precharge_heap.append('1')
471         Addr_0_heap.append('0')
472         Addr_1_heap.append('0')
473         Addr_2_heap.append('0')
474         Addr_3_heap.append('0')
475         Addr_4_heap.append('0')
476         write_array(FileOut2,[str(count)]+command)
477     elif last[0] == 'ADD':
478       for i in range(10):
479         count = count + 1
480         command[9] = last[6]
481         command[10] = last[2]
482         command[11] = last[4]
483         command[12] = command[13] = command[14] = command[16] = command[17] =
      command[18] = command[19] = command[27] = command[28] ='0'
484         command[15] = command[29] = '1'
485         command[20] = command[21] = '0'
486         command[22] = '1'
487         ## Get value from heap
488         command[24] = Write_en_heap[0]
489         Write_en_heap = Write_en_heap[1:]
490         command[25] = Read_en_heap[0]
491         Read_en_heap = Read_en_heap[1:]
492         command[26] = Precharge_heap[0]
493         Precharge_heap = Precharge_heap[1:]
494         command[4] = Addr_0_heap[0]
495         Addr_0_heap = Addr_0_heap[1:]
496         command[5] = Addr_1_heap[0]
497         Addr_1_heap = Addr_1_heap[1:]
498         command[6] = Addr_2_heap[0]
499         Addr_2_heap = Addr_2_heap[1:]
500         command[7] = Addr_3_heap[0]
501         Addr_3_heap = Addr_3_heap[1:]
502         command[8] = Addr_4_heap[0]
503         Addr_4_heap = Addr_4_heap[1:]
504         ## Insert value to heap
505         Write_en_heap.append('0')
506         Read_en_heap.append('0')
507         Precharge_heap.append('1')
508         Addr_0_heap.append('0')
509         Addr_1_heap.append('0')
510         Addr_2_heap.append('0')
511         Addr_3_heap.append('0')
512         Addr_4_heap.append('0')
513         write_array(FileOut2,[str(count)]+command)
```

```
514   elif last[0] == 'ADDI':
515     for i in range(10):
516       count = count + 1
517       command[0] = last[4][0]
518       command[1] = last[4][1]
519       command[2] = last[4][2]
520       command[3] = last[4][3]
521       command[9] = last[6]
522       command[10] = last[2]
523       command[13] = command[14] = command[16] = command[17] = command[18] =
      command[19] = command[27] = command[28] ='0'
524       command[12] = command[15] = command[29] = '1'
525       command[20] = command[21] = '0'
526       command[22] = '1'
527       ## Get value from heap
528       command[24] = Write_en_heap[0]
529       Write_en_heap = Write_en_heap[1:]
530       command[25] = Read_en_heap[0]
531       Read_en_heap = Read_en_heap[1:]
532       command[26] = Precharge_heap[0]
533       Precharge_heap = Precharge_heap[1:]
534       command[4] = Addr_0_heap[0]
535       Addr_0_heap = Addr_0_heap[1:]
536       command[5] = Addr_1_heap[0]
537       Addr_1_heap = Addr_1_heap[1:]
538       command[6] = Addr_2_heap[0]
539       Addr_2_heap = Addr_2_heap[1:]
540       command[7] = Addr_3_heap[0]
541       Addr_3_heap = Addr_3_heap[1:]
542       command[8] = Addr_4_heap[0]
543       Addr_4_heap = Addr_4_heap[1:]
544       ## Insert value to heap
545       Write_en_heap.append('0')
546       Read_en_heap.append('0')
547       Precharge_heap.append('1')
548       Addr_0_heap.append('0')
549       Addr_1_heap.append('0')
550       Addr_2_heap.append('0')
551       Addr_3_heap.append('0')
552       Addr_4_heap.append('0')
553       write_array(FileOut2,[str(count)]+command)
554   elif last[0] == 'MUL':
555     for i in range(10):
556       count = count + 1
557       command[9] = last[6]
558       command[10] = last[2]
559       command[11] = last[4]
560       command[12] = command[13] = command[14] = command[15] = command[16] =
      command[18] = command[19] = command[27] = command[28] ='0'
561       command[17] = command[29] = '1'
562       command[20] = '0'
563       command[21] = command[22] = '1'
564       ## Get value from heap
565       command[24] = Write_en_heap[0]
566       Write_en_heap = Write_en_heap[1:]
567       command[25] = Read_en_heap[0]
```

```
568         Read_en_heap = Read_en_heap[1:]
569         command[26] = Precharge_heap[0]
570         Precharge_heap = Precharge_heap[1:]
571         command[4] = Addr_0_heap[0]
572         Addr_0_heap = Addr_0_heap[1:]
573         command[5] = Addr_1_heap[0]
574         Addr_1_heap = Addr_1_heap[1:]
575         command[6] = Addr_2_heap[0]
576         Addr_2_heap = Addr_2_heap[1:]
577         command[7] = Addr_3_heap[0]
578         Addr_3_heap = Addr_3_heap[1:]
579         command[8] = Addr_4_heap[0]
580         Addr_4_heap = Addr_4_heap[1:]
581         ## Insert value to heap
582         Write_en_heap.append('0')
583         Read_en_heap.append('0')
584         Precharge_heap.append('1')
585         Addr_0_heap.append('0')
586         Addr_1_heap.append('0')
587         Addr_2_heap.append('0')
588         Addr_3_heap.append('0')
589         Addr_4_heap.append('0')
590         write_array(FileOut2,[str(count)]+command)
591     elif last[0] == 'MULI':
592       for i in range(10):
593         count = count + 1
594         lalala = Four_bit(last[4])
595         command[0] = lalala[0]
596         command[1] = lalala[1]
597         command[2] = lalala[2]
598         command[3] = lalala[3]
599         command[9] = last[6]
600         command[10] = last[2]
601         command[13] = command[14] = command[15] = command[16] = command[18] =
      command[19] = command[27] = command[28] ='0'
602         command[12] = command[17] = command[29] = '1'
603         command[20] = '0'
604         command[21] = command[22] = '1'
605         ## Get value from heap
606         command[24] = Write_en_heap[0]
607         Write_en_heap = Write_en_heap[1:]
608         command[25] = Read_en_heap[0]
609         Read_en_heap = Read_en_heap[1:]
610         command[26] = Precharge_heap[0]
611         Precharge_heap = Precharge_heap[1:]
612         command[4] = Addr_0_heap[0]
613         Addr_0_heap = Addr_0_heap[1:]
614         command[5] = Addr_1_heap[0]
615         Addr_1_heap = Addr_1_heap[1:]
616         command[6] = Addr_2_heap[0]
617         Addr_2_heap = Addr_2_heap[1:]
618         command[7] = Addr_3_heap[0]
619         Addr_3_heap = Addr_3_heap[1:]
620         command[8] = Addr_4_heap[0]
621         Addr_4_heap = Addr_4_heap[1:]
622         ## Insert value to heap
```

```
623         Write_en_heap.append('0')
624         Read_en_heap.append('0')
625         Precharge_heap.append('1')
626         Addr_0_heap.append('0')
627         Addr_1_heap.append('0')
628         Addr_2_heap.append('0')
629         Addr_3_heap.append('0')
630         Addr_4_heap.append('0')
631         write_array(FileOut2,[str(count)]+command)
632     elif last[0] == 'MIN':
633       for i in range(10):
634         count = count + 1
635         command[9] = last[6]
636         command[10] = last[2]
637         command[11] = last[4]
638         command[12] = command[13] = command[14] = command[15] = command[17] =
      command[18] = command[19] = command[27] = command[28] ='0'
639         command[16] = command[29] = '1'
640         command[20] = command[22] = '0'
641         command[21] = '1'
642         ## Get value from heap
643         command[24] = Write_en_heap[0]
644         Write_en_heap = Write_en_heap[1:]
645         command[25] = Read_en_heap[0]
646         Read_en_heap = Read_en_heap[1:]
647         command[26] = Precharge_heap[0]
648         Precharge_heap = Precharge_heap[1:]
649         command[4] = Addr_0_heap[0]
650         Addr_0_heap = Addr_0_heap[1:]
651         command[5] = Addr_1_heap[0]
652         Addr_1_heap = Addr_1_heap[1:]
653         command[6] = Addr_2_heap[0]
654         Addr_2_heap = Addr_2_heap[1:]
655         command[7] = Addr_3_heap[0]
656         Addr_3_heap = Addr_3_heap[1:]
657         command[8] = Addr_4_heap[0]
658         Addr_4_heap = Addr_4_heap[1:]
659         ## Insert value to heap
660         Write_en_heap.append('0')
661         Read_en_heap.append('0')
662         Precharge_heap.append('1')
663         Addr_0_heap.append('0')
664         Addr_1_heap.append('0')
665         Addr_2_heap.append('0')
666         Addr_3_heap.append('0')
667         Addr_4_heap.append('0')
668         write_array(FileOut2,[str(count)]+command)
669     elif last[0] == 'MINI':
670       for i in range(10):
671         count = count + 1
672         command[0] = last[4][0]
673         command[1] = last[4][1]
674         command[2] = last[4][2]
675         command[3] = last[4][3]
676         command[9] = last[6]
677         command[10] = last[2]
```

```
678         command[13] = command[14] = command[15] = command[17] = command[18] =
    command[19] = command[27] = command[28] ='0'
679         command[12] = command[16] = command[29] = '1'
680         command[20] = command[22] = '0'
681         command[21] = '1'
682         ## Get value from heap
683         command[24] = Write_en_heap[0]
684         Write_en_heap = Write_en_heap[1:]
685         command[25] = Read_en_heap[0]
686         Read_en_heap = Read_en_heap[1:]
687         command[26] = Precharge_heap[0]
688         Precharge_heap = Precharge_heap[1:]
689         command[4] = Addr_0_heap[0]
690         Addr_0_heap = Addr_0_heap[1:]
691         command[5] = Addr_1_heap[0]
692         Addr_1_heap = Addr_1_heap[1:]
693         command[6] = Addr_2_heap[0]
694         Addr_2_heap = Addr_2_heap[1:]
695         command[7] = Addr_3_heap[0]
696         Addr_3_heap = Addr_3_heap[1:]
697         command[8] = Addr_4_heap[0]
698         Addr_4_heap = Addr_4_heap[1:]
699         ## Insert value to heap
700         Write_en_heap.append('0')
701         Read_en_heap.append('0')
702         Precharge_heap.append('1')
703         Addr_0_heap.append('0')
704         Addr_1_heap.append('0')
705         Addr_2_heap.append('0')
706         Addr_3_heap.append('0')
707         Addr_4_heap.append('0')
708         write_array(FileOut2,[str(count)]+command)
709     elif last[0] == 'SFL':
710       for i in range(10):
711         count = count + 1
712         command[0] = last[4][0]
713         command[1] = last[4][1]
714         command[2] = last[4][2]
715         command[3] = last[4][3]
716         command[9] = last[6]
717         command[10] = last[2]
718         command[13] = command[14] = command[15] = command[16] = command[17] =
    command[19] = command[27] = command[28] ='0'
719         command[12] = command[18] = command[29] = '1'
720         command[21] = command[22] = '0'
721         command[20] = '1'
722         ## Get value from heap
723         command[24] = Write_en_heap[0]
724         Write_en_heap = Write_en_heap[1:]
725         command[25] = Read_en_heap[0]
726         Read_en_heap = Read_en_heap[1:]
727         command[26] = Precharge_heap[0]
728         Precharge_heap = Precharge_heap[1:]
729         command[4] = Addr_0_heap[0]
730         Addr_0_heap = Addr_0_heap[1:]
731         command[5] = Addr_1_heap[0]
```

```
732         Addr_1_heap = Addr_1_heap[1:]
733         command[6] = Addr_2_heap[0]
734         Addr_2_heap = Addr_2_heap[1:]
735         command[7] = Addr_3_heap[0]
736         Addr_3_heap = Addr_3_heap[1:]
737         command[8] = Addr_4_heap[0]
738         Addr_4_heap = Addr_4_heap[1:]
739         ## Insert value to heap
740         Write_en_heap.append('0')
741         Read_en_heap.append('0')
742         Precharge_heap.append('1')
743         Addr_0_heap.append('0')
744         Addr_1_heap.append('0')
745         Addr_2_heap.append('0')
746         Addr_3_heap.append('0')
747         Addr_4_heap.append('0')
748         write_array(FileOut2,[str(count)]+command)
749     elif last[0] == 'SFR':
750       for i in range(10):
751         count = count + 1
752         command[0] = last[4][0]
753         command[1] = last[4][1]
754         command[2] = last[4][2]
755         command[3] = last[4][3]
756         command[9] = last[6]
757         command[10] = last[2]
758         command[13] = command[14] = command[15] = command[16] = command[17] =
      command[27] = command[28] ='0'
759         command[12] = command[18] = command[19] = command[29] = '1'
760         command[21] = command[22] = '0'
761         command[20] = '1'
762         ## Get value from heap
763         command[24] = Write_en_heap[0]
764         Write_en_heap = Write_en_heap[1:]
765         command[25] = Read_en_heap[0]
766         Read_en_heap = Read_en_heap[1:]
767         command[26] = Precharge_heap[0]
768         Precharge_heap = Precharge_heap[1:]
769         command[4] = Addr_0_heap[0]
770         Addr_0_heap = Addr_0_heap[1:]
771         command[5] = Addr_1_heap[0]
772         Addr_1_heap = Addr_1_heap[1:]
773         command[6] = Addr_2_heap[0]
774         Addr_2_heap = Addr_2_heap[1:]
775         command[7] = Addr_3_heap[0]
776         Addr_3_heap = Addr_3_heap[1:]
777         command[8] = Addr_4_heap[0]
778         Addr_4_heap = Addr_4_heap[1:]
779         ## Insert value to heap
780         Write_en_heap.append('0')
781         Read_en_heap.append('0')
782         Precharge_heap.append('1')
783         Addr_0_heap.append('0')
784         Addr_1_heap.append('0')
785         Addr_2_heap.append('0')
786         Addr_3_heap.append('0')
```

```
787         Addr_4_heap.append('0')
788         write_array(FileOut2,[str(count)]+command)
```

Code Listing 1.16: Part 3: Vector File Generation

### 1.6.2 Back-End Part

Here shows the back-end Python code so that we can use it to compare the result and see if CPU is operating properly.

```
1  import numpy as np
2  import random
3  zeros = '0000000000000000'
4  def or_op(a,b):
5    return bin(int(a,2) | int(b,2))[2:].zfill(16)
6  def and_op(a,b):
7    return bin(int(a,2) & int(b,2))[2:].zfill(16)
8  def add_op(a,b):
9    return bin(int(a,2) + int(b,2))[2:].zfill(16)
10 def min_op(a,b):
11   return bin(min(int(a,2) , int(b,2)))[2:].zfill(16)
12 def mul_op(a,b):
13   return bin((int(a[11:],2) * int(b[11:],2)))[2:].zfill(16)
14 def sfl_op(a,b):
15   return (a[int(b,2):])+zeros[0:int(b,2)]
16 def sfr_op(a,b):
17   return a[0:len(a)-int(b,2)].zfill(16)
18
19 FileIn = open('output.txt','r')
20 FileOut = open('result.txt','w')
21 mem = ['0' for i in range(32)]
22 reg = ['0' for i in range(8)]
23 for i in FileIn:
24   ins = i.split()
25   if ins[0] == 'STOREI':
26     mem[int(ins[-1],2)]=bin(int(ins[2],16))[2:].zfill(16)
27   if ins[0] == 'STORE':
28     mem[int(ins[-1],2)]=reg[int(ins[2])]
29   if ins[0] == 'LOADI':
30     reg[int(ins[-1])]=bin(int(ins[2],16))[2:].zfill(16)
31   if ins[0] == 'LOAD':
32     reg[int(ins[-1])]=mem[int(ins[2],2)]
33   if ins[0] == 'MULI':
34     reg[int(ins[-1])] = mul_op(reg[int(ins[2])],bin(int(ins[4],16))[2:].zfill
       (16))
35   if ins[0] == 'MUL':
36     reg[int(ins[-1])] = mul_op(reg[int(ins[2])],reg[int(ins[4])])
37   if ins[0] == 'ANDI':
38     reg[int(ins[-1])] = and_op(reg[int(ins[2])],bin(int(ins[4],16))[2:])
39   if ins[0] == 'AND':
40     reg[int(ins[-1])] = and_op(reg[int(ins[2])],reg[int(ins[4])])
41   if ins[0] == 'ORI':
42     reg[int(ins[-1])] = or_op(reg[int(ins[2])],bin(int(ins[4],16))[2:])
43   if ins[0] == 'OR':
44     reg[int(ins[-1])] = or_op(reg[int(ins[2])],reg[int(ins[4])])
45   if ins[0] == 'ADDI':
```

```
46      reg[int(ins[-1])] = add_op(reg[int(ins[2])],bin(int(ins[4],16))[2:])
47    if ins[0] == 'ADD':
48      reg[int(ins[-1])] = add_op(reg[int(ins[2])],reg[int(ins[4])])
49    if ins[0] == 'MINI':
50      reg[int(ins[-1])] = min_op(reg[int(ins[2])],bin(int(ins[4],16))[2:])
51    if ins[0] == 'MIN':
52      reg[int(ins[-1])] = min_op(reg[int(ins[2])],reg[int(ins[4])])
53    if ins[0] == 'SFL':
54      reg[int(ins[-1])] = sfl_op(reg[int(ins[2])],bin(int(ins[4],16))[2:])
55    if ins[0] == 'SFR':
56      reg[int(ins[-1])] = sfr_op(reg[int(ins[2])],bin(int(ins[4],16))[2:])
57
58  result_dec = [str(int(i,2)) for i in reg]
59  result_hex = [str(hex(int(i, 2))[2:].zfill(4)) for i in reg]
60  FileOut.write('\t'+'Reg0'+'\t'+'Reg1'+'\t'+'Reg2'+'\t'+'Reg3'+'\t'+'Reg4'+'\t'+
        'Reg5'+'\t'+'Reg6'+'\t'+'Reg7'+'\n')
61  FileOut.write('HEX'+'\t'+result_hex[0]+'\t'+result_hex[1]+'\t'+result_hex[2]+'\
        t'+result_hex[3]+'\t'+result_hex[4]+'\t'+result_hex[5]+'\t'+result_hex[6]+'
        \t'+result_hex[7]+'\n')
62  FileOut.write('DEC'+'\t'+result_dec[0]+'\t'+result_dec[1]+'\t'+result_dec[2]+'\
        t'+result_dec[3]+'\t'+result_dec[4]+'\t'+result_dec[5]+'\t'+result_dec[6]+'
        \t'+result_dec[7]+'\n')
63  print result_hex
64  print result_dec
```

Code Listing 1.17: Back-End Python Code

# 2. Bayesian Neural Network (BNN)

The goal of this project is to get exposed to a research-oriented topic and leverage what we have learned during the course to address it. In this project, we implement the BNN Accelerator to fulfill a specific problem, which calculate the multiplication of 5 pairs of X and W and give the summation of the 5 items. We first read some papers providing some background on Bayesian Neural Networks (BNNs) and Gaussian random number generators (GRNGs). Then compare two GRNG architectures and find that Wallace Method suits best our hardware accelerator design. Finally, we implement the GRNG using python and the BNN using cadence virtuoso tool. Report the performance of our design.

## 2.1 Schematic Design

This part show the design of our BNN circuit. Though we could send 5 pairs of X and W to the circuit at one time and sum them up in the following pipelined stage, but the overhead, both layout and timing, of this design will be huge as the result that we need to build the BNN circuit with 5 multipliers blocks and 5 adders. Therefore, we came up with a better plan for realizing this functionality of our BNN. The following structure shows the basic concept of our design. Since we keep reusing the same block every clock, our clock frequency will be 2.5ns and it will give us final result around 26ns. However, in order to compare with our general purpose CPU, we change the clock frequency to 5ns. Then we will get final result around 43ns.

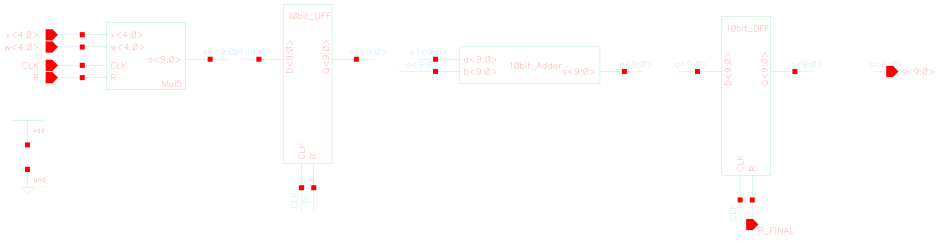

**Figure 2.1:** *BNN schematic*

## 2.2 Layout Design

Since we use a simple architecture in this part, both the area and time will be reduced. The following part will show the layout of our BNN design.

**Figure 2.2:** *BNN layout*

## 2.3 LVS Result

The following part show the LVS report of our BNN extract file.

```
@(#)$CDS: LVS version 6.1.7-64b 07/05/2016 20:10 (sjfhw313) $

Command line: /usr/local/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /
    home/scf-16/shangzho/cds/LVS -l -s -t /home/scf-16/shangzho/cds/LVS/layout
    /home/scf-16/shangzho/cds/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/scf-16/shangzho/cds/LVS/layout/netlist
       count
        1254            nets
        25              terminals
```

```
        2409            pmos
        2409            nmos

    Net-list summary for /home/scf-16/shangzho/cds/LVS/schematic/netlist
       count
        1254            nets
        25              terminals
        1369            pmos
        1369            nmos


    Terminal correspondence points
    N1240      N50        CLK
    N1236      N49        R
    N1231      N26        R_FINAL
    N1233      N1         gnd!
    N1239      N21        s<0>
    N1235      N14        s<1>
    N1230      N35        s<2>
    N1252      N28        s<3>
    N1250      N46        s<4>
    N1248      N45        s<5>
    N1246      N15        s<6>
    N1244      N19        s<7>
    N1241      N6         s<8>
    N1237      N47        s<9>
    N1243      N0         vdd!
    N1245      N11        w<0>
    N1242      N2         w<1>
    N1238      N4         w<2>
    N1234      N9         w<3>
    N1229      N10        w<4>
    N1232      N16        x<0>
    N1253      N23        x<1>
    N1251      N30        x<2>
    N1249      N29        x<3>
    N1247      N51        x<4>

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   0       0
        active                   4818    2738
        total                    4818    2738

                                   nets
        un-matched               0       0
```

```
        merged                  0       0
        pruned                  0       0
        active                  1254    1254
        total                   1254    1254

                                terminals
        un-matched              0       0
        matched but
        different type          0       0
        total                   25      25


Probe files from /home/scf-16/shangzho/cds/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/scf-16/shangzho/cds/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

Code Listing 2.1: BNN LVS

## 2.4 Python Code List

Here shows the Python code for BNN Wallace-GRNG.

```
import math
import numpy
import random

```

```
## Define positive decimal to binary by divide-by-2 method
def convert(dec):
  if (dec == 0):
    return ''
  else:
    ## Recursion the function
    return convert(dec//2)+str(dec%2)

## Define decimal to binary function
def dec_to_bin(input_dec):
  # Negative case
  if (input_dec < 0):
    temp = input_dec + 32
  ## Positive case
  else:
    temp = input_dec
  a = convert(temp)
  ## Add bit for small positive output
  if (len(a)<5):
    for i in range(5-len(a)):
      a='0'+a
  return a

def matrix_mul(A,B):
  c = [[0],[0],[0],[0]]
  for i in range (4):
    for j in range (4):
      c[i][0] = c[i][0]+A[i][j]*B[j][0]
  return c

def pool_generate(N):
  pool = [[],[],[],[]]
  count = 0
  ## Generate a 4x1024 size pool
  s = [random.randint(-15,15) for i in range(N)]
  for j in range(L):
    for z in range(K):
      pool[z].append(s[count])
      count = count+1
  return pool

## Apply wallace method
def wallace(pool,R,L,K,A):
  for i in range (R):
    for j in range (L):
      for z in range (K):
        x[z][0] = pool[z][j]
      ## Apply matrix transformation to the K values
      x_bar = matrix_mul(A,x)
      ## write K values back to pool
      for z in range (K):
        pool[z][j] = x_bar[z][0]
  ## Apply correction for sum of squares
  S = 1
  ## Return pool with scaled sum of squares
  for j in range (L):
```

```
61      for z in range (K):
62        pool[z][j] = pool[z][j]*S
63    #pool[K-1][L-1] = pool[K-1][L-1]/S
64    return pool
65
66  ## Shift order of pool
67  def GRNG_shift(pool):
68    shifted = []
69    shifted_new = []
70    for i in range(L):
71      for j in range(K):
72        shifted.append(pool[j][i])
73    shifted_new.append(shifted[-1])
74    for i in range(len(shifted)-1):
75      shifted_new.append(shifted[i])
76    return shifted_new
77
78  ## Initialization
79  ## Hadamard Matrix
80  A=[[-0.5,0.5,0.5,0.5], [0.5,-0.5,0.5,0.5], [-0.5,-0.5,0.5,-0.5],
        [-0.5,-0.5,-0.5,0.5]]
81  x = [[0],[0],[0],[0]]
82  ## Retention Factor
83  R = 2
84  ## N/K
85  L = 1024
86  ## Matrix Size
87  K = 4
88  ## Variational Parameters
89  mu = [1,1,3,3,2]
90  sigma = [1,2,3,2,1]
91  W = []
92  ebsilon = []
93  ebsilon_bin = []
94  ## Generate pool
95  pool = pool_generate(K*L)
96  ## Apply Wallace method
97  pool_new = wallace(pool,R,L,K,A)
98  ## Apply shifting
99  pool_shift = GRNG_shift(pool_new)
100 ## Generate weights
101 for i in range(5):
102   ## Apply weight generation
103   temp = int(pool_shift[random.randint(0,K*L)])
104   ebsilon.append(temp)
105   ebsilon_bin.append(dec_to_bin(temp))
106   W.append(temp*mu[i]+sigma[i])
107 print "W =",W
108 print "Epsilon =",ebsilon
109 print "Epsilon_BIN =",ebsilon_bin
```

Code Listing 2.2: BNN Wallace-GRNG

The code can be divided into two part. First part is the definitions of functions, and second part is for generating the weight values in the for loop. Here shows the five generated weight according to the variational parameters $\mu$ and $\sigma$.

# 3. Simulation Result

## 3.1 CPU Functionality Simulation

### 3.1.1 Input Instructions List

Here shows the input instructions to test the functionality of the CPU.

```
STOREI 0AH #002f
STOREI 0BH #0004
STOREI 2 10H #000B #00EE
LOAD $1 0AH
LOAD $2 0BH
LOAD $3 10H
LOAD $4 11H

MUL $5 $1 $2
MUL $6 $3 $4
NOP
NOP
STORE 00H $5
STORE 01H $6

SFL $5 $3 #0003
OR $6 $5 $4
ANDI $6 $6 #00AA
ADD $6 $6 $4
STORE 02H $5
STORE 03H $6

LOAD $1 00H
LOAD $2 01H
LOAD $3 02H
LOAD $4 03H
```

Code Listing 3.1: ISA Input Instructions Listing

### 3.1.2 Virtuoso Simulation Result

Here shows the Virtuoso simulation result of CPU with ISA input. The input vector file is as below:

```
radix 4 4 4 4 1 1 1 1 1 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
io   i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i
```

```
vname value_in<[15:12]> value_in<[11:8]> value_in<[7:4]> value_in<[3:0]>
    instr_addr_in<4> instr_addr_in<3> instr_addr_in<2> instr_addr_in<1>
    instr_addr_in<0> Dest_reg<[2:0]> read_sel1<[2:0]> read_sel2<[2:0]>
    VRselect ANDen OREn ADDen SUBen MULen Shiften Shiftdir Destvalue<2>
    Destvalue<1> Destvalue<0> Data_in write_en read_en pre Value_select
    Mem_select Reg_write
vih 1.8
slope 0.001
tunit 0.5ns
1 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
2 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
3 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
4 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
5 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
6 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
7 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
8 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
9 0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
10  0 0 2 f 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
11  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
12  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
13  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
14  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
15  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
16  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
17  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
18  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
19  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
20  0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
21  0 0 0 B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
22  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
23  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
24  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
25  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
26  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
27  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
28  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
29  0 0 0 B 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
30  0 0 0 B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
31  0 0 E E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
32  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
33  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
34  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
35  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
36  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
37  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
38  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
39  0 0 E E 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0
40  0 0 E E 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
41  0 0 E E 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1
42  0 0 E E 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1
43  0 0 E E 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1
44  0 0 E E 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1
45  0 0 E E 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1
46  0 0 E E 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 1
47  0 0 E E 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 1
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 48 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 49 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 50 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 51 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 52 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 53 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 54 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 55 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 56 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 57 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 58 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 59 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 60 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 61 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 62 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 63 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 64 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 65 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 66 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 67 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 68 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 69 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 70 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 71 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 72 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 73 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 74 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 75 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 76 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 77 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 78 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 79 | 0 | 0 | E | E | 0 | 1 | 0 | 1 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 80 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 81 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 82 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 83 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 84 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 85 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 86 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 87 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 88 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 89 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 90 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 91 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 92 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 93 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 94 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 95 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 96 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 97 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 98 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 99 | 0 | 0 | E | E | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 100 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 101 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 102 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 103 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

```
104 0 0 E E 0 0 0 0 0 5 1 2 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
105 0 0 E E 0 0 0 0 0 5 1 2 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
106 0 0 E E 0 0 0 0 0 5 1 2 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
107 0 0 E E 0 0 0 0 0 5 1 2 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
108 0 0 E E 0 0 0 0 0 5 1 2 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
109 0 0 E E 0 0 0 0 0 5 1 2 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
110 0 0 E E 0 0 0 0 0 5 1 2 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
111 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
112 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
113 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
114 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
115 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
116 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
117 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
118 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
119 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
120 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1
121 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
122 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
123 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
124 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
125 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
126 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
127 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
128 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
129 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
130 0 0 E E 0 0 0 0 0 6 3 4 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
131 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
132 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
133 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
134 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
135 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
136 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
137 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
138 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
139 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
140 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
141 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
142 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
143 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
144 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
145 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
146 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
147 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
148 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
149 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
150 0 0 E E 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
151 0 0 E E 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
152 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
153 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
154 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
155 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
156 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0
157 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0
158 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0
159 0 0 E E 0 1 1 0 0 6 6 4 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 160 | 0 | 0 | E | E | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 161 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 162 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 163 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 164 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 165 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 166 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 167 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 168 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 169 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 170 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 171 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 172 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 173 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 174 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 175 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 176 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 177 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 178 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 179 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 180 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 181 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 182 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 183 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 184 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 185 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 186 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 187 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 188 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 189 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 190 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 191 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 192 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 193 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 194 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 195 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 196 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 197 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 198 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 199 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 200 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 201 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 202 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 203 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 204 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 205 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 206 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 207 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 208 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 209 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 210 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 211 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 212 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 213 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 214 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 215 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

```
216 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
217 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
218 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
219 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
220 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
221 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
222 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
223 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
224 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
225 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
226 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
227 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
228 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
229 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
230 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
231 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
232 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
233 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
234 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
235 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
236 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
237 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
238 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
239 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
240 0 0 0 3 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0
241 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
242 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
243 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
244 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
245 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
246 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
247 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
248 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
249 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
250 0 0 A A 0 0 0 0 0 6 6 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
251 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
252 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
253 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
254 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
255 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
256 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
257 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
258 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
259 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
260 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
261 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
262 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
263 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
264 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
265 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
266 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
267 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
268 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
269 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
270 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
271 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
```

```
272 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
273 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
274 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
275 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
276 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
277 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
278 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
279 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
280 0 0 A A 0 0 0 0 0 6 6 4 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
281 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
282 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
283 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
284 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
285 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
286 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
287 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
288 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
289 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
290 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
291 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
292 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
293 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
294 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
295 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
296 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
297 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
298 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
299 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
300 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
301 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
302 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
303 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
304 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
305 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
306 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
307 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
308 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
309 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
310 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
311 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
312 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
313 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
314 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
315 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
316 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0
317 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0
318 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0
319 0 0 A A 0 0 0 1 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0
320 0 0 A A 0 0 0 0 0 6 5 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
321 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
322 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
323 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
324 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
325 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
326 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
327 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
```

```
328 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
329 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
330 0 0 A A 0 0 0 0 0 6 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
331 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
332 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
333 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
334 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
335 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
336 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
337 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
338 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
339 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
340 0 0 A A 0 0 0 0 0 1 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
341 0 0 A A 0 0 0 0 0 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
342 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
343 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
344 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
345 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
346 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1
347 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1
348 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1
349 0 0 A A 0 0 0 1 1 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1
350 0 0 A A 0 0 0 0 0 2 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
351 0 0 A A 0 0 0 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
352 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
353 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
354 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
355 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
356 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
357 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
358 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
359 0 0 A A 0 1 1 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
360 0 0 A A 0 0 0 0 0 3 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
361 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
362 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
363 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
364 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
365 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
366 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
367 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
368 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
369 0 0 A A 0 0 0 0 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1
370 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1
371 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0
372 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0
373 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
374 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
375 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
376 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
377 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
378 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
379 0 0 A A 0 0 0 1 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
380 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
381 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0
382 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0
383 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
```

```
384 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
385 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
386 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
387 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
388 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
389 0 0 A A 0 0 0 1 1 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0
390 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
391 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
392 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
393 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
394 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
395 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
396 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
397 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
398 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
399 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
400 0 0 A A 0 0 0 0 0 4 6 4 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0
```

Code Listing 3.2: ISA Vector File Listing

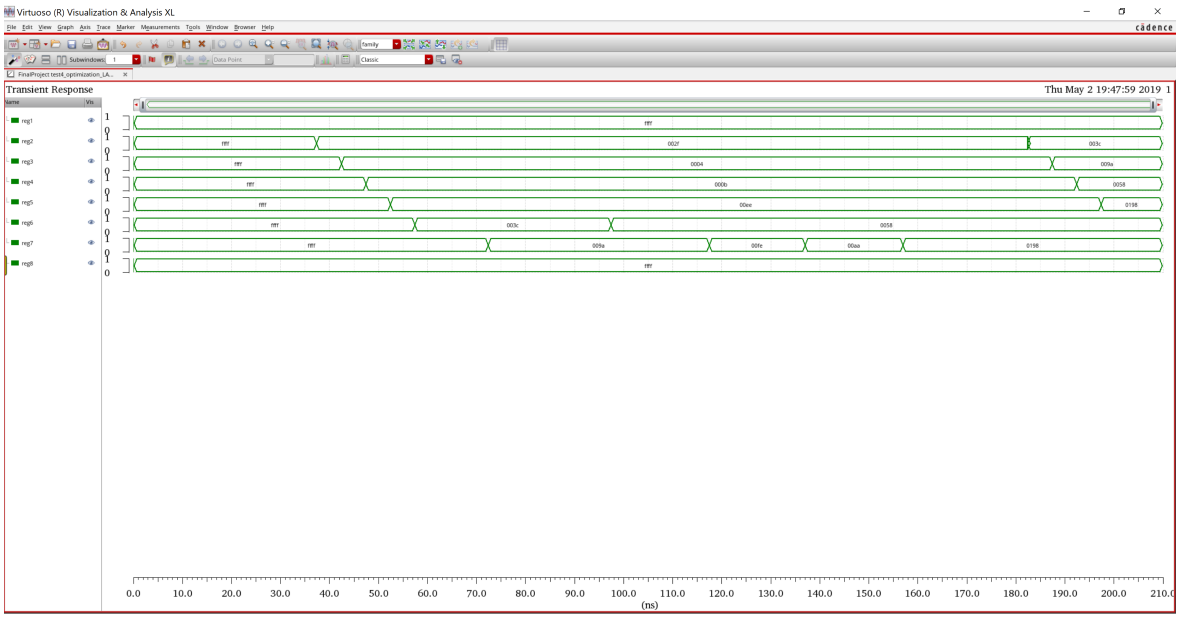


**Figure 3.1:** *Virtuoso Simulation Result of CPU with ISA Input*

The data saved in 8 registers are given. Since in instructions, Reg $1 and Reg $7 are not used, there values are set randomly all the way, here they are 'ffff'.

### 3.1.3 Back-End Result

Here shows the back-end result generated from the Python code.

```
  Reg0   Reg1   Reg2   Reg3   Reg4   Reg5   Reg6   Reg7
HEX 0000   003c   009a   0058   0198   0058   0198   0000
DEC 0  60   154  88   408  88   408  0
```

Code Listing 3.3: ISA Result from Back-End Code

The result generated from the back-end code can be matched with the simulation result from Virsuoso, which means the system is functioning properly.

## 3.2 BNN Functionality Simulation

### 3.2.1 Output of BNN Based GRNG

The section below shows the output, some Gaussian random numbers generated from the Python code given in previous chapters.

```
W = [-5, -4, 18, 11, 29]
Epsilon = [-6, -6, 5, 3, 14]
Epsilon_BIN = ['11010', '11010', '00101', '00011', '01110']
```

Code Listing 3.4: Test Result of BNN Wallace-GRNG

The given variational parameters $\mu$, $\sigma$ and the calculated weights are:

$$\mu = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 3 \\ 2 \end{bmatrix} \quad \sigma = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 1 \end{bmatrix} \quad \epsilon = \begin{bmatrix} -6 \\ -6 \\ 5 \\ 3 \\ 14 \end{bmatrix} \quad \epsilon_{BIN} = \begin{bmatrix} 11010 \\ 11010 \\ 00101 \\ 00011 \\ 01110 \end{bmatrix} \quad W = \begin{bmatrix} -5 \\ -4 \\ 18 \\ 11 \\ 29 \end{bmatrix}$$

### 3.2.2 Simulation Result

For BNN part, we build both the schematic and layout circuits. The 5 weights and inputs are: $\text{W}= \begin{bmatrix} 00101 \\ 01000 \\ 00011 \\ 00000 \\ 00001 \end{bmatrix}$ $\text{X}= \begin{bmatrix} 00010 \\ 01011 \\ 00111 \\ 01110 \\ 00100 \end{bmatrix}$

#### Schematic Simulation

Here shows the simulation result of BNN schematic circuit.

#### Layout Simulation

Here shows the simulation result of BNN layout circuit.

### 3.2.3 Back-End Result

Here shows the back-end result generated from the Python code.

```
  Reg0   Reg1   Reg2   Reg3   Reg4   Reg5   Reg6   Reg7
HEX 007b   0077   0058   0015   0000   0004   0000   0000
DEC 123  119  88   21   0  4  0  0
```

Code Listing 3.5: BNN Result from Back-End Code

The result generated from the back-end code can be matched with the simulation result from Virsuoso, which means the system is functioning properly.


**Figure 3.2:** *Virtuoso Simulation Result of BNN Schematic*

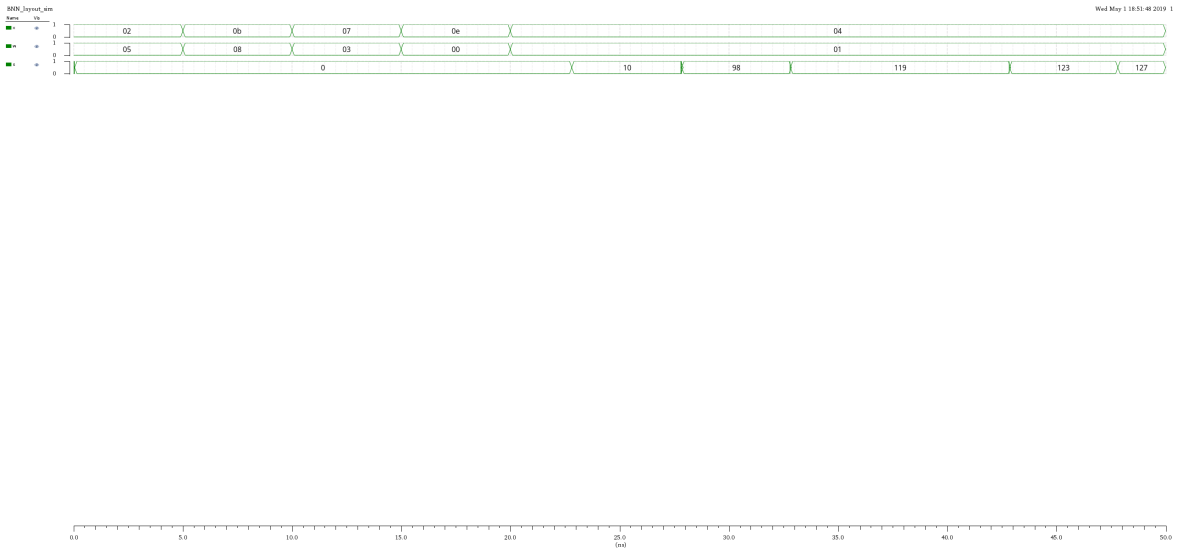


**Figure 3.3:** *Virtuoso Simulation Result of BNN Layout*

## 3.3 CPU & BNN Performance Comparison

To compare the performance of CPU and BNN, the simulation of CPU given the same input to BNN is shown here.

The input instructions given to CPU is as:

```
  Reg0   Reg1   Reg2   Reg3   Reg4   Reg5   Reg6   Reg7
HEX 007b   0077   0058   0015   0000   0004   0000   0000
DEC 123  119  88   21   0  4  0  0
```

Code Listing 3.6: BNN Test Instruction for CPU

The corresponding vector file is:

```
radix 4 4 4 4 1 1 1 1 1 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
io   i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i
vname value_in<[15:12]> value_in<[11:8]>  value_in<[7:4]> value_in<[3:0]>
    instr_addr_in<4>  instr_addr_in<3>  instr_addr_in<2>  instr_addr_in<1>
    instr_addr_in<0>  Dest_reg<[2:0]> read_sel1<[2:0]>  read_sel2<[2:0]>
    VRselect  ANDen OREn  ADDen SUBen MULen Shiften Shiftdir  Destvalue<2>
    Destvalue<1>  Destvalue<0>  Data_in write_en  read_en pre Value_select
    Mem_select  Reg_write
vih 1.8
slope 0.001
tunit 0.5ns
1 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
2 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
3 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
4 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
5 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
6 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
7 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
8 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
9 0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
10  0 0 0 5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
11  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
12  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
13  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
14  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
15  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
16  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
17  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
18  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
19  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
20  0 0 0 8 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
21  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
22  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
23  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
24  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
25  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
26  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
27  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
28  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
29  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
30  0 0 0 3 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
31  0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
32  0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 41 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 42 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 43 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 44 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 45 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 46 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 47 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 48 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 49 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 50 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 51 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 52 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 53 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 54 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 55 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 56 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 57 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 58 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 59 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 60 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 61 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 62 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 63 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 64 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 65 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 66 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 67 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 68 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 69 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 70 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 71 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 72 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 73 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 74 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 75 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 76 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 77 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 78 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 79 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 80 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 81 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 82 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 83 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 84 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 85 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 86 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 87 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 88 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 89 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 90 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 91 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 92 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 93 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 94 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 95 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 96 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 97 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 98 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 99 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 100 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 101 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 102 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 103 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 104 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 105 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 106 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 107 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 108 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 109 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 110 | 0 | 0 | 0 | B | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 111 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 112 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 113 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 114 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 115 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 116 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 117 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 118 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 119 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 120 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 121 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 122 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 123 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 124 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 125 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 126 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 127 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 128 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 129 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 130 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 131 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 132 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 133 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 134 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 135 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 136 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 137 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 138 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 139 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 140 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 141 | 0 | 0 | 0 | E | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 142 | 0 | 0 | 0 | E | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 143 | 0 | 0 | 0 | E | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 144 | 0 | 0 | 0 | E | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

```
145 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
146 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
147 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
148 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
149 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
150 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
151 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
152 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
153 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
154 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
155 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
156 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
157 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
158 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
159 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
160 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
161 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
162 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
163 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
164 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
165 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
166 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
167 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
168 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
169 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
170 0 0 0 E 0 0 0 0 0 4 4 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
171 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
172 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
173 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
174 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
175 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
176 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
177 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
178 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
179 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
180 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1
181 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
182 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
183 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
184 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
185 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
186 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
187 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
188 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
189 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
190 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
191 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
192 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
193 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
194 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
195 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
196 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
197 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
198 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
199 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
200 0 0 0 4 0 0 0 0 0 5 5 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
```

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 201 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 202 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 203 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 204 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 205 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 206 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 207 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 208 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 209 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 210 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 211 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 212 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 213 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 214 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 215 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 216 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 217 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 218 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 219 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 220 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 221 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 222 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 223 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 224 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 225 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 226 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 227 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 228 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 229 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 230 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 231 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 232 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 233 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 234 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 235 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 236 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 237 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 238 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 239 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 240 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 241 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 242 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 243 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 244 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 245 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 246 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 247 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 248 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 249 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 250 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 251 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 252 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 253 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 254 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 255 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 256 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

```
257 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
258 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
259 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
260 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
261 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
262 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
263 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
264 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
265 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
266 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
267 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
268 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
269 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
270 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
271 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
272 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
273 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
274 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
275 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
276 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
277 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
278 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
279 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
280 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
281 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
282 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
283 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
284 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
285 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
286 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
287 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
288 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
289 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
290 0 0 0 4 0 0 0 0 0 1 1 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
291 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
292 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
293 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
294 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
295 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
296 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
297 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
298 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
299 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
300 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1
301 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
302 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
303 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
304 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
305 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
306 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
307 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
308 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
309 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
310 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
311 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
312 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
```

```
313 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
314 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
315 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
316 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
317 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
318 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
319 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
320 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
321 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
322 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
323 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
324 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
325 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
326 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
327 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
328 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
329 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
330 0 0 0 4 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
```

Code Listing 3.7: BNN Test Instruction for CPU

### 3.3.1 Simulation result

Here shows the simulation result of CPU with BNN case input.

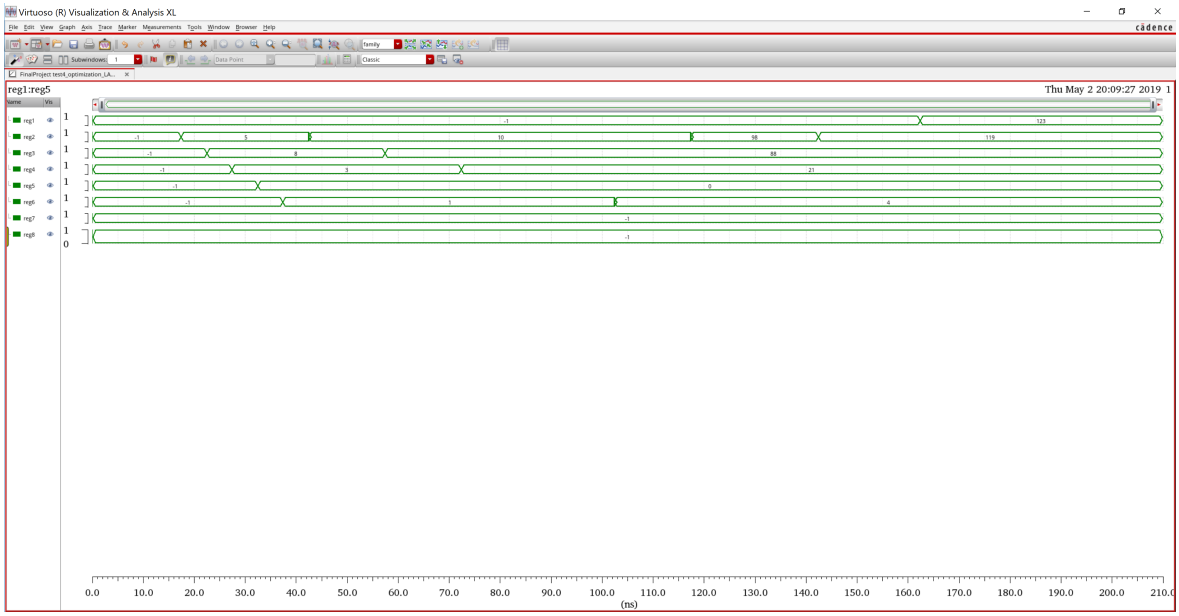


**Figure 3.4:** *Virtuoso Simulation Result of CPU with BNN Input*

The final result given by CPU is the same with the output of BNN circuit. The time that CPU circuit spending to get the final result is about 160 ns, however, for BNN circuit, it's only about 40 ns.

The conclusion is that the CPU circuit can be used to processing multi-task including "AND", "OR" and other functions, while BNN circuit only has the specific function, which is

to operate multiplication and addition in a specific way. Therefore, circuit which has a more fixed structure will have a better performance since there is less unused circuit which may cause delay.

## 3.4 CPU Parameter Conclusion

**Table 3.1:** *CPU Parameter Conclusion*

| **Clock** ($ns$) | **Area** ($\mu m^2$) | **Power** ($mW$) | **Clock**$\times$ **Area**$\times$ **Power** ($ns \times \mu m^2 \times mW$) |
|---|---|---|---|
| 5.0 | 145,325 | 3.84 | 2,790,240 |

# 4. Conclusion

In this Final project, we apply what we learned in 577a into real practice. Also, we use clock-gating, dynamic logic, register re-balancing and TG FF to optimize our CPU. For the BNN optimization, we use a simple circuit which reuse our adder and multiplier inside the BNN circuit. For the scripting part, we are getting familiar with the python scripting language by write the front-end code to decode given ISA instructions. Moreover, we also complete the back-end python test code for testing our results coming from the CPU. It turns out the result of our CPU for a set of instructions is correct. We also come to know the benefits of Out-of-Order (OoO) execution as it can give a higher throughput of our pipelined CPU. Furthermore, there could be a promising optimization for this OoO execution which could be done in our python script. As we stated in phase II, we accomplished our OoO execution for MUL/MULI instruction by re-arrange the instruction sequence in our python code rather than blindly insert NOP, which is meaningless for outputs. If we did this, our throughput will increase at the same time. The speed up for the BNN circuit is 3.7X comparing to our general purpose CPU design. Therefore, if you want to do a specific operation such as multiplication and addition, BNN circuit will be the best choice compared to the general purpose CPU.

Finally, we would like to express our deep gratitude to Professor Pierluigi and TAs, for their patient guidance, enthusiastic encouragement and useful critiques of this project work.

# References

[1] KillaKem. Subtraction using adder circuit, 2014.

[2] Greg Novick Crystal Chen and Kirk Shimano. RISC Architecture-How Pipelining Works, 2000.

[3] Neeta Pandey and Saurabh Gupta. Design and implementation of novel multiplier using barrel shifters. *International Journal of Image, Graphics and Signal Processing*, 7:28–34, 07 2015.