Advanced Computer-Aided VLSI System Design

Homework 1: Run-length Encoder with Low Power Design

Graduate Institute of Electronics Engineering, National Taiwan University

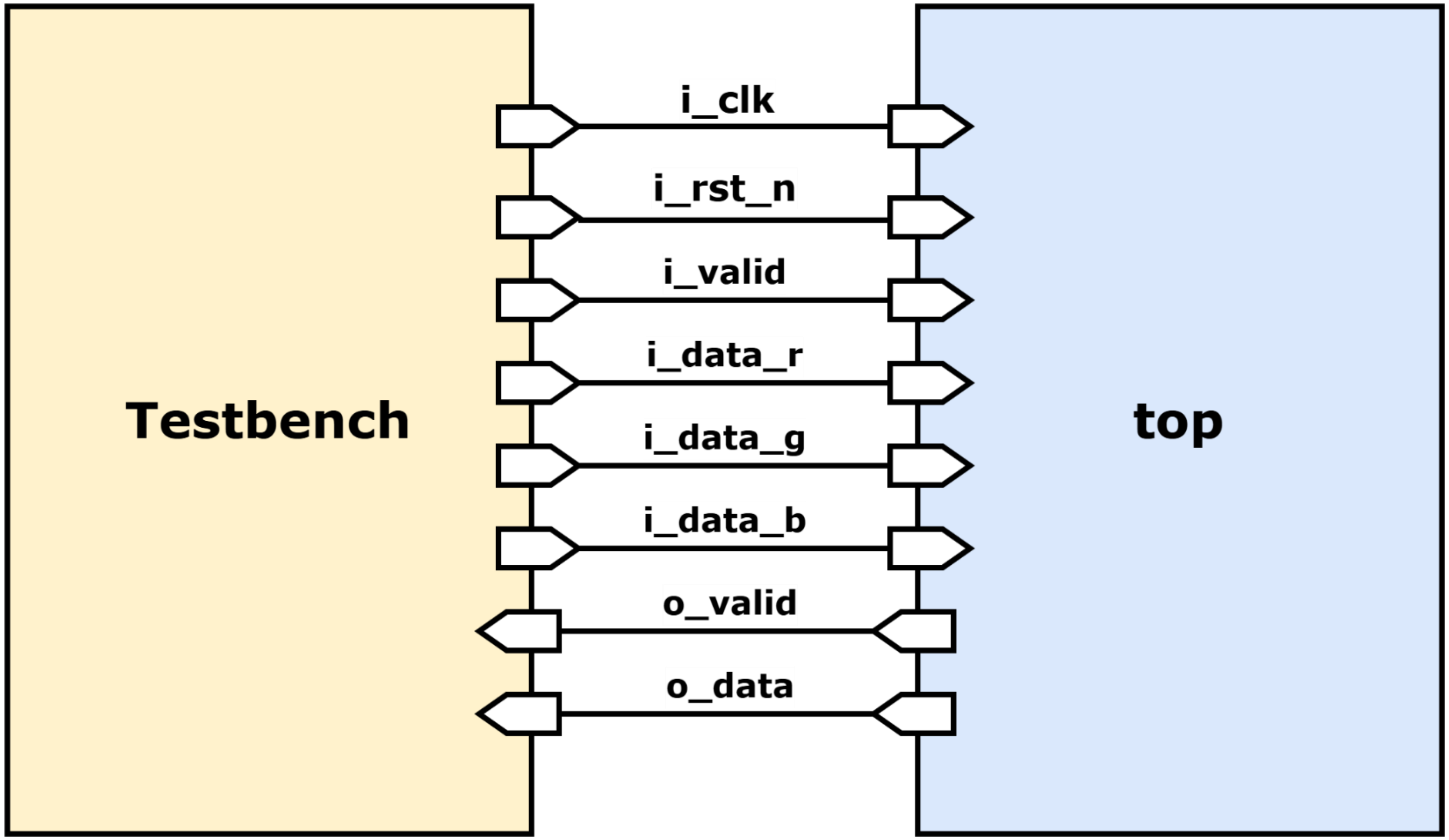
Goal

- In this homework, you will learn
  - The flow of the low-power solution
  - Write UPF file to control power domain
  - Implement a low-power design constraint through power reduction methods
  - Simulation commands with VCS NLP
  - Use primetime to estimate power

Introduction

- In this homework, you are going to design a simplified Run-length Encoder and implement it with low-power design constraint through UPF flow

Block Diagram



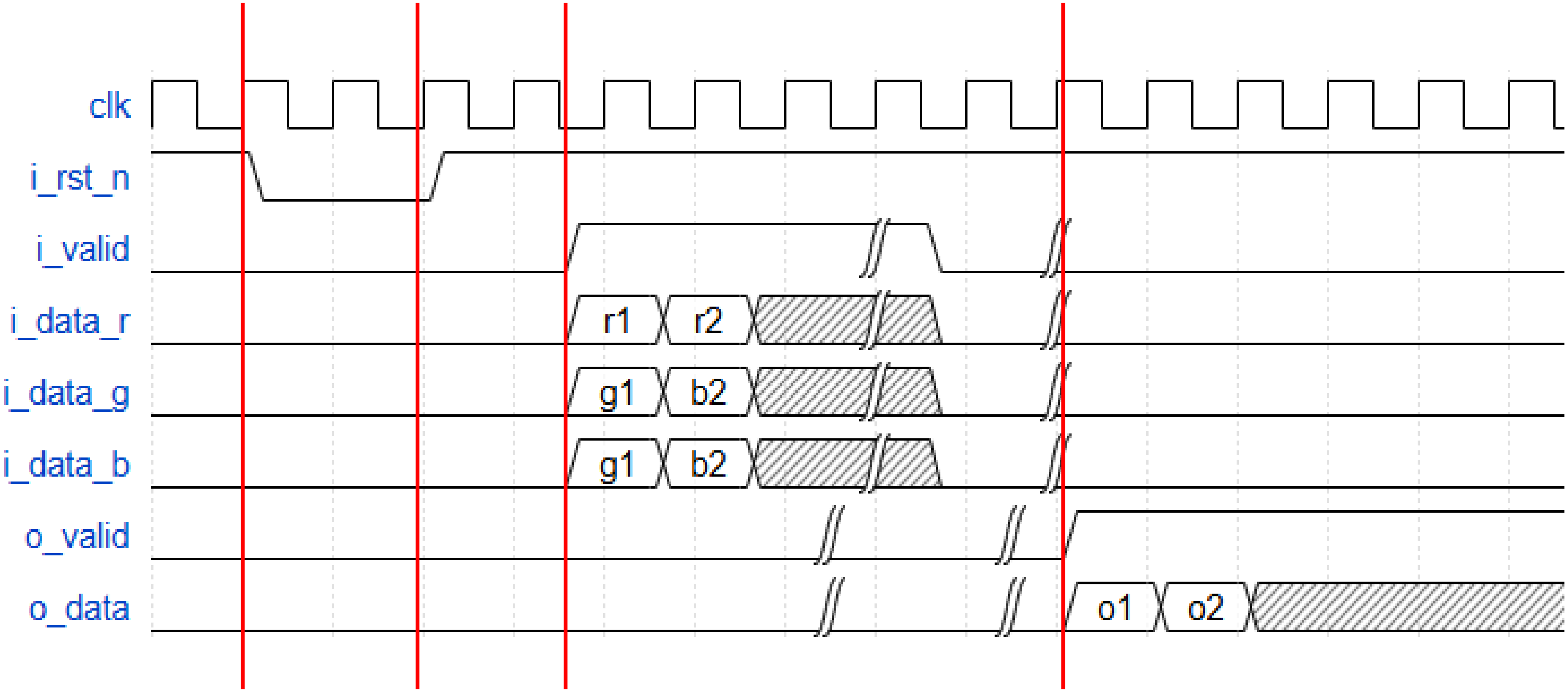
Input/Output

Signal Name	I/O	Width	Simple Description
i_clk	I	1	Clock signal in the system
i_rst_n	I	1	Active **low** asynchronous reset
i_valid	I	1	The signal is **high** if input data is ready
i_data_r	I	8	Unsigned integer input data which represent the red, green, blue value of each pixel from 0 to 255
i_data_g	I	8	
i_data_b	I	8	
o_valid	O	1	Set **high** if ready to output result
o_data	O	18	Unsigned output data after run-length encoding 1. For o_data[17:10], output the pixel value of each run 2. For o_data[9:0], output the length of the value

Specification (1/2)

- Active low asynchronous reset is used only once
- All inputs are synchronized with the **negative** clock edge
- All outputs should be synchronized with the **positive** clock edge
- New pattern (i_data_r, i_data_g and i_data_b) is ready only when i_valid is high
- o_valid should be high for only one cycle for each o_data
- The testbench will sample o_data at **negative** clock edge if o_valid is high
- You can raise o_valid at any moment **but only once**

Specification (2/2)



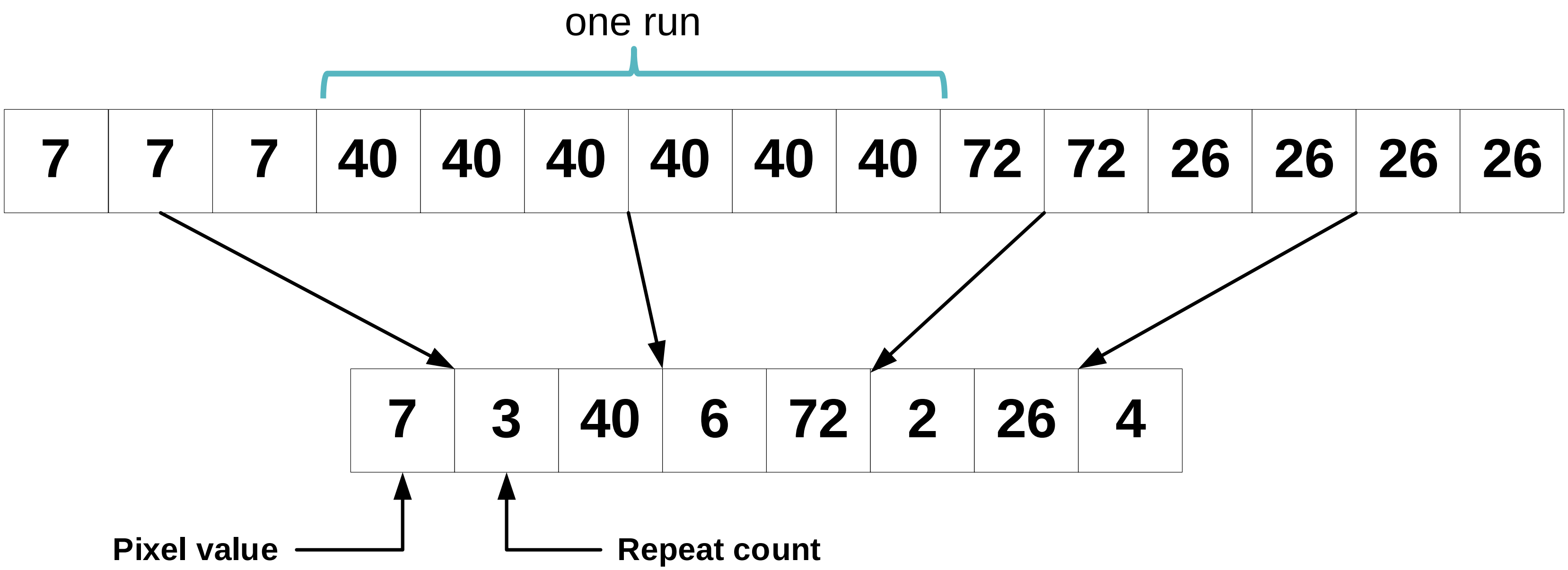
Reset signal

Sending input patterns

Output patterns

Run-length Encoding (1/2)

- Runs of data (consecutive occurrences of the same data value) are stored as a single occurrence
  - Includes data value and a count of its consecutive occurrences

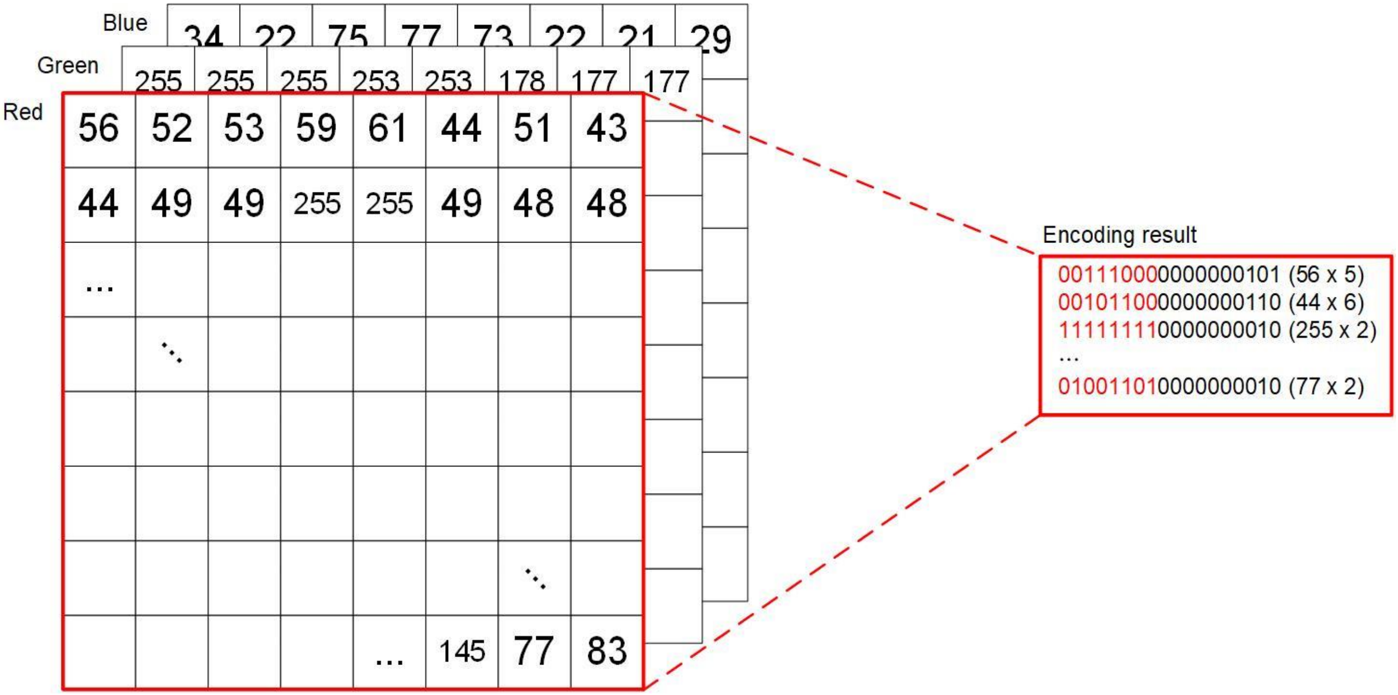


7	7	7	40	40	40	40	40	40	72	72	26	26	26	26

7	3	40	6	72	2	26	4

Run-length Encoding (2/2)

- Lossy run-length encoding
  - One value is first picked as the values reference value
  - Pixels between a certain range, instead of the exact same one, are stored in one run

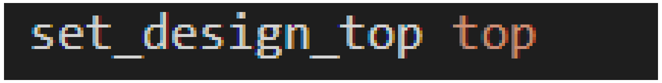


Encoding result

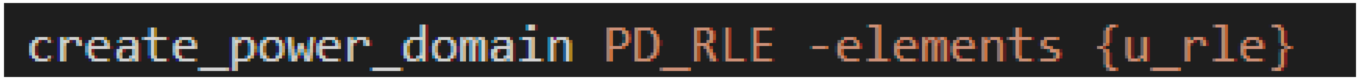
```
001110000000000101 (56 x 5)
001011000000000110 (44 x 6)
111111110000000010 (255 x 2)
...
010011010000000010 (77 x 2)
```

Requirements for UPF

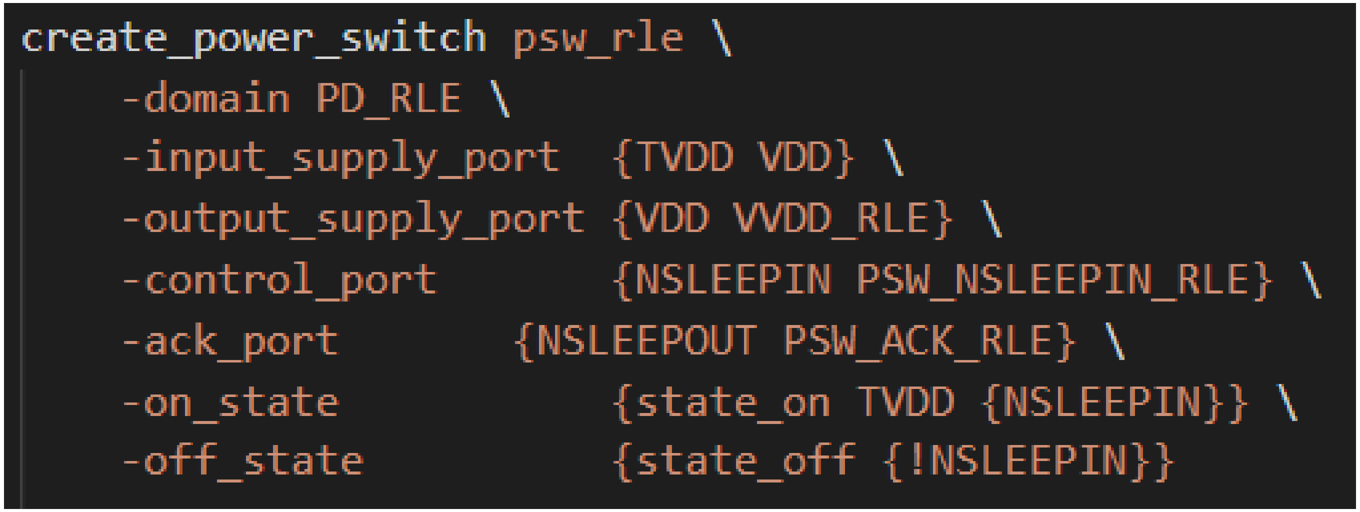
- Power top should be set in the UPF file
- At least **two power domains** (including top domain) are used in your design
- At least **one power switch** is used in your design



```
set_design_top top
```

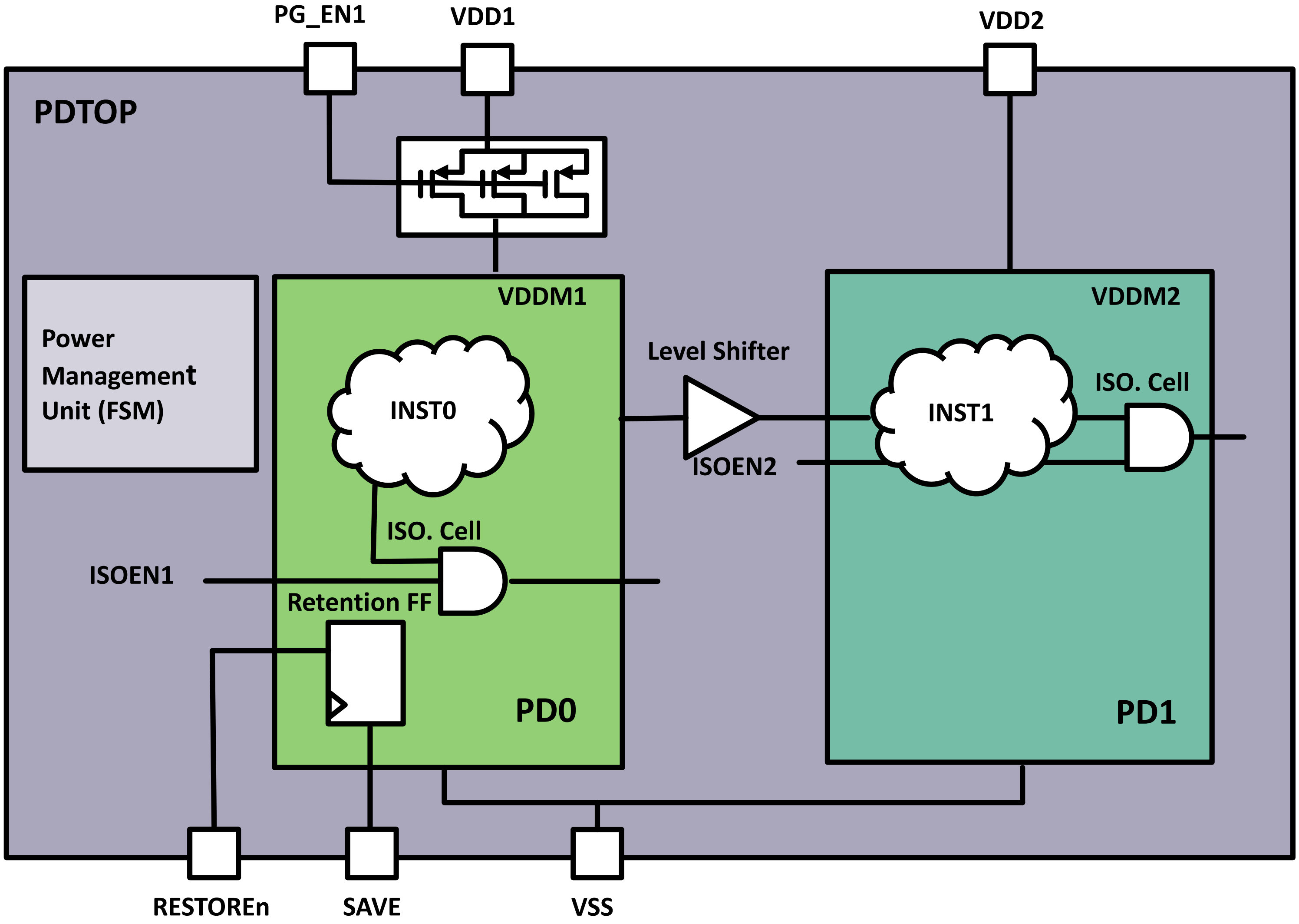


```
create_power_domain PD_RLE -elements {u_rle}
```



```
create_power_switch psw_rle \
    -domain PD_RLE \
    -input_supply_port  {TVDD VDD} \
    -output_supply_port {VDD VVDD_RLE} \
    -control_port       {NSLEEPIN PSW_NSLEEPIN_RLE} \
    -ack_port       {NSLEEPOUT PSW_ACK_RLE} \
    -on_state           {state_on TVDD {NSLEEPIN}} \
    -off_state          {state_off {!NSLEEPIN}}
```

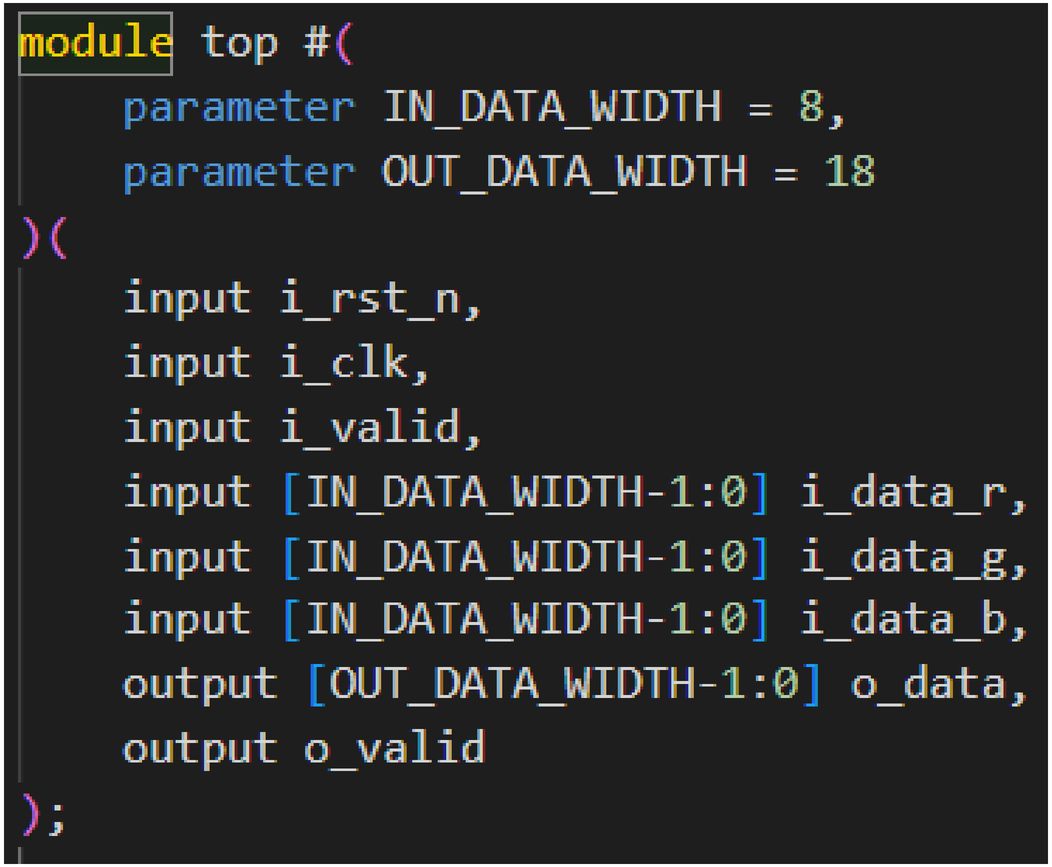
Reference for Low Power Strategy



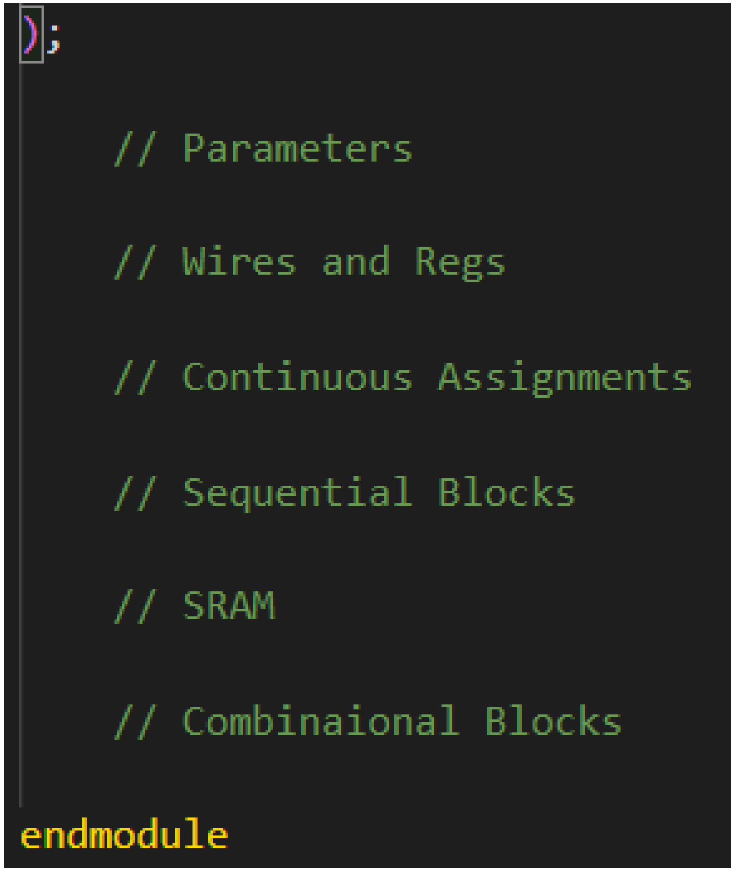
Other Requirements

- The range threshold value is set to **10**
- The sizes of all input images are 64x64x3 pixels
- You are **NOT** allowed to use DesignWare

top.v



```
module top #(
    parameter IN_DATA_WIDTH = 8,
    parameter OUT_DATA_WIDTH = 18
)(
    input i_rst_n,
    input i_clk,
    input i_valid,
    input [IN_DATA_WIDTH-1:0] i_data_r,
    input [IN_DATA_WIDTH-1:0] i_data_g,
    input [IN_DATA_WIDTH-1:0] i_data_b,
    output [OUT_DATA_WIDTH-1:0] o_data,
    output o_valid
);
```



```
);

    // Parameters

    // Wires and Regs

    // Continuous Assignments

    // Sequential Blocks

    // SRAM

    // Combinaional Blocks

endmodule
```

N16 ADFP Memory

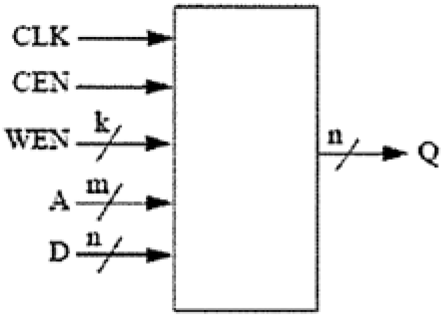
- No memory compiler in N16 ADFP
- Can only use the provided SRAM

TS1 (single port)	TS6 (two port)
TS1N16ADFPCLLLVTA16X88M2SWSHOD	TS6N16ADFPCLLLVTA16X32M2FWSHOD
TS1N16ADFPCLLLVTA16X96M2SWSHOD	TS6N16ADFPCLLLVTA16X72M2FWSHOD
TS1N16ADFPCLLLVTA128X64M4SWSHOD	TS6N16ADFPCLLLVTA16X120M2FWSHOD
TS1N16ADFPCLLLVTA512X45M4SWSHOD	TS6N16ADFPCLLLVTA32X32M2FWSHOD
	TS6N16ADFPCLLLVTA128X32M4FWSHOD
	TS6N16ADFPCLLLVTA128X64M4FWSHOD

Single Port Memory IO

- CEB: active low chip enable
- WEB: write: 1’b0, read: 1’b1
- DSLP/SLP: (Deep) Sleep
- SD: Shut Down

```
module TS1N16ADFPCLLLVTA128X64M4SWSHOD(
    .A        (address),
    .CEB      (cen),   // active low
    .CLK      (clk),
    .WEB      (wen),   // write: low, read: high
    .D        (datain),
    .Q        (dataout),
    .BWEB     ({64{1’b0}}),
    .RTSEL    (2’b01),
    .WTSEL    (2’b01),
    .SLP      (1’b0),
    .DSLP     (1’b0),
    .SD       (1’b0),
    .PUDELAY  ()
);
```

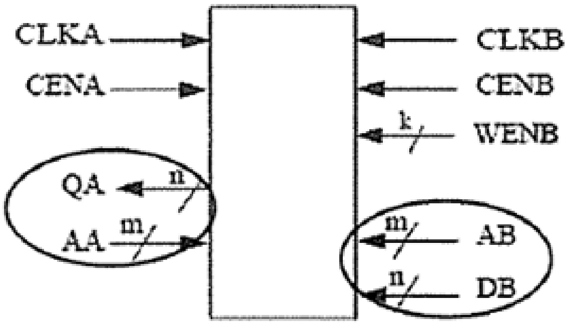


Single Port Memory

Two Port Memory IO

- REB: active low read enable
- WEB: active low write enable

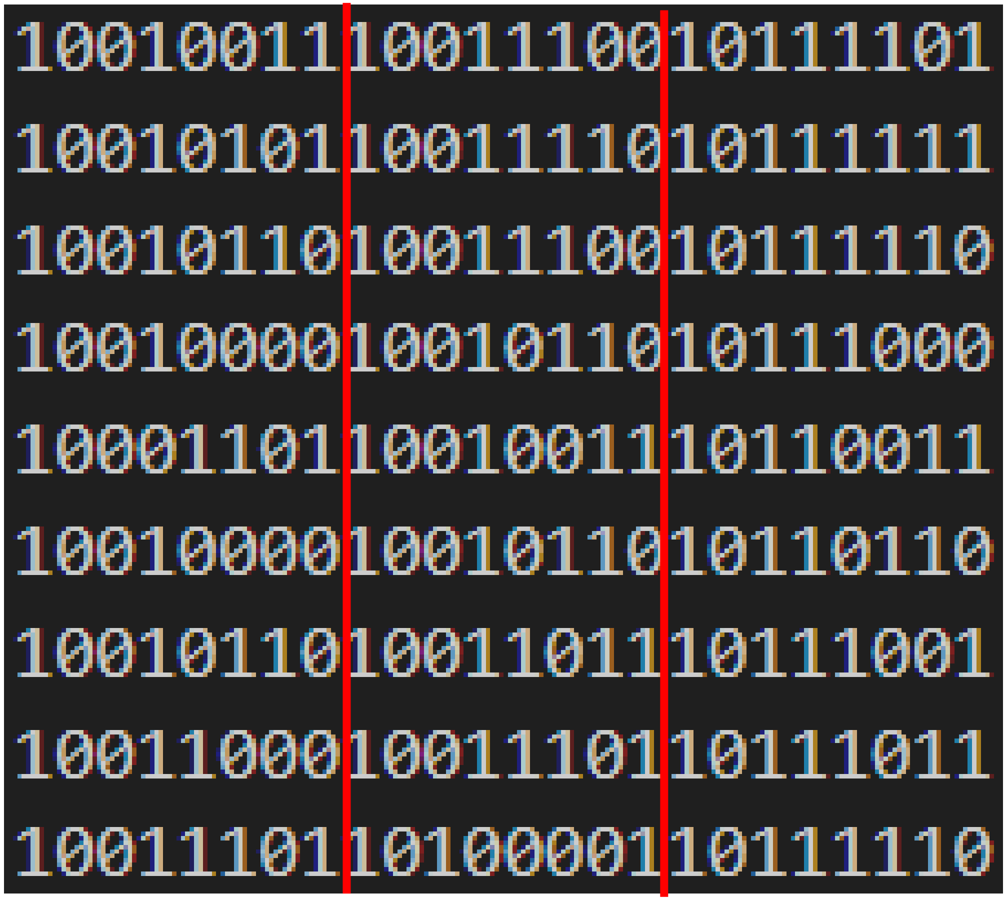
```
module TS6N16ADFPCLLLVTA128X32M4FWSHOD(
    .AB       (read_address),
    .REB      (read_en),     // active low
    .CLKR     (read_clk),
    .Q        (dataout),
    .AA       (write_address),
    .WEB      (write_en),    // active low
    .D        (datain),
    .CLKW     (write_clk),
    .BWEB     ({32{1’b0}}),
    .RCT      (2’b01),
    .WCT      (2’b01),
    .KP       (3’b011),
    .SLP      (1’b0),
    .DSLP     (1’b0),
    .SD       (1’b0),
    .PUDELAY ()
);
```



Two Port Memory

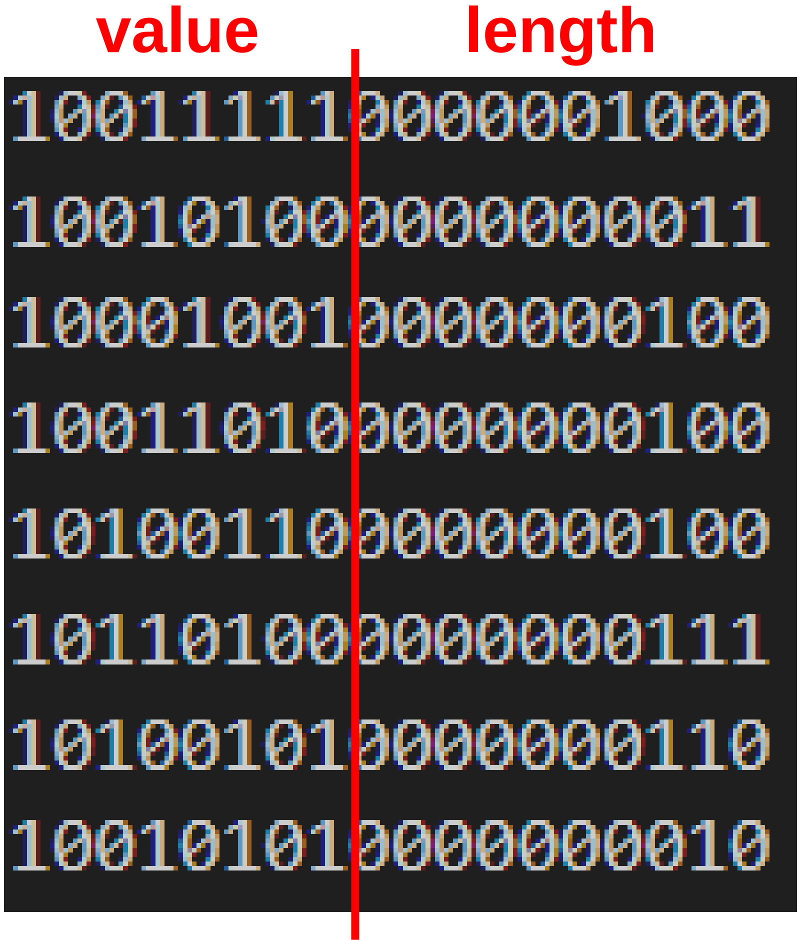
Pattern (Input Data)

i_data_r i_data_g i_data_b



```
100100111001110010111101
100101011001111010111111
100101101001110010111110
100100001001011010111000
100011011001001110110011
100100001001011010110110
100101101001101110111001
100110001001110110111011
100111011010000110111110
```

Pattern (Golden Output)

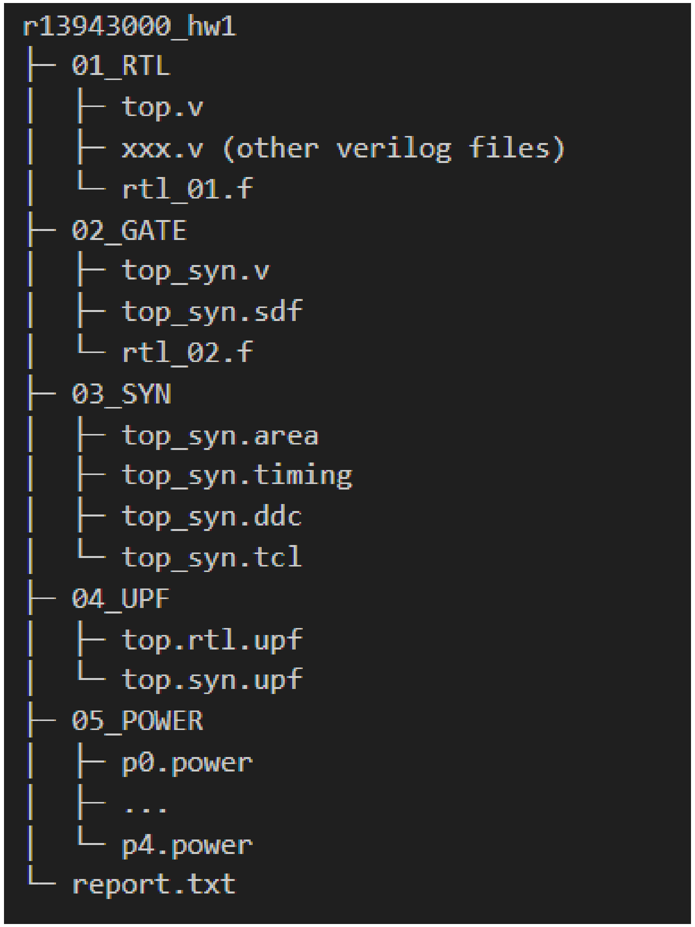


value length

```
100111110000001000
100101000000000011
100010010000000100
100110100000000100
101001100000000100
101101000000000111
101001010000000110
100101010000000010
```

Submission

- Create a folder named **studentID_hw1** and follow the hierarchy below (*.sv is allowed if you use SystemVerilog)



```
r13943000_hw1
├─ 01_RTL
│   ├─ top.v
│   ├─ xxx.v (other verilog files)
│   └─ rtl_01.f
├─ 02_GATE
│   ├─ top_syn.v
│   ├─ top_syn.sdf
│   └─ rtl_02.f
├─ 03_SYN
│   ├─ top_syn.area
│   ├─ top_syn.timing
│   ├─ top_syn.ddc
│   └─ top_syn.tcl
├─ 04_UPF
│   ├─ top.rtl.upf
│   └─ top.syn.upf
├─ 05_POWER
│   ├─ p0.power
│   ├─ ...
│   └─ p4.power
└─ report.txt
```

Submission - Workstation

- Pack the folder **studentID_hw1** into a **tar.gz file** named **acvsdxxx-hw1.tar.gz**
  - **tar -zcvf acvsdxxx-hw1.tar.gz [path to studentID_hw1]**
  - Use **lowercase** for all the letters (e.g. acvsd000-hw1.tar.gz)
  - Pack the folder on ADFP server to avoid OS related problems
  - Place the tar.gz file at the root of your ADFP account

- TA will only check the last version after the homework deadline
- Reminder
  - Files uploaded to ADFP workstation must be kept as a copy in your local folder or computer

Submission - NTU COOL

- For design files and UPF scripts (01_RTL, 04_UPF, report)
- Pack the folder **studentID_hw1** into a **tar file** named **acvsdxxx-hw1-vk.tar.gz** (*k* is the number of version, *k* =1,2,…)
  - **tar -zcvf acvsdxxx-hw1-vk.tar.gz [path to studentID_hw1]**
  - Use **lowercase** for all the letters. (e.g. acvsd000-hw1-v1.tar.gz)
  - Pack the folder on IC Design LAB server to avoid OS related problems
- Submit to NTU Cool

Grading Policy (1/3)

- Grading command for RTL

  -
    ```
    vcs -full64 -R -f rtl_01.f +v2k -sverilog -debug_access+all
    +define+$1
    ```

  -
    ```
    vcs -full64 -R -f rtl_01.f –upf [path to UPF file] +v2k -
    sverilog -debug_access+all +define+$1+UPF
    ```
- Simulation: **70%**
  - Need to pass all of 5 public and 5 private test cases

	Score
RTL simulation	15%
RTL with UPF	15%
Gate-level simulation	20%
Gate-level with UPF	20%

Grading Policy (2/3)

- Grading command for gate-level

  -
    ```
    vcs -full64 -R -f rtl_02.f +v2k-sverilog -v2005 -debug_access+all +maxdelays -negdelay +neg_tchk +define+$1+SDF
    ```

  -
    ```
    vcs -full64 -R -f rtl_02.f –upf [path to UPF file] +v2k -sverilog -v2005 -debug_access+all +maxdelays -negdelay +neg_tchk +define+$1+SDF+UPF
    ```
- Performance: **30%**
  - Use public test cases (But need to pass hidden pattern)
  - Performance = $\sum(P^2 \times T) \times A$
    - P: Power (mW), T: Simulation time (ns), A: Area (um^2)
  - Baseline = 3×10^6

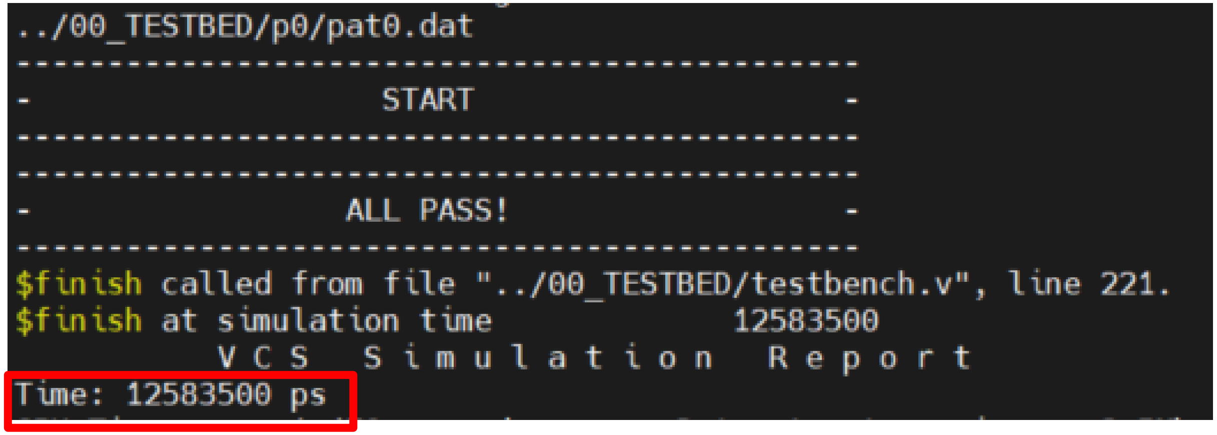
	Score
Baseline	10%
Ranking (Need to pass baseline)	20%

Grading Policy (3/3)

- Lose 5 points for any incorrect naming or format
- Violations of any spec incur point penalties
  - Negative slack/Design without SRAM
    - 0 point for gate-level simulations and performance
  - Violate other rules but pass all simulations
    - Performance score * 0.7
- No late submission
  - **0 point** for this homework
- No plagiarism
  - Plagiarism in any form, including copying from online sources, is strictly prohibited

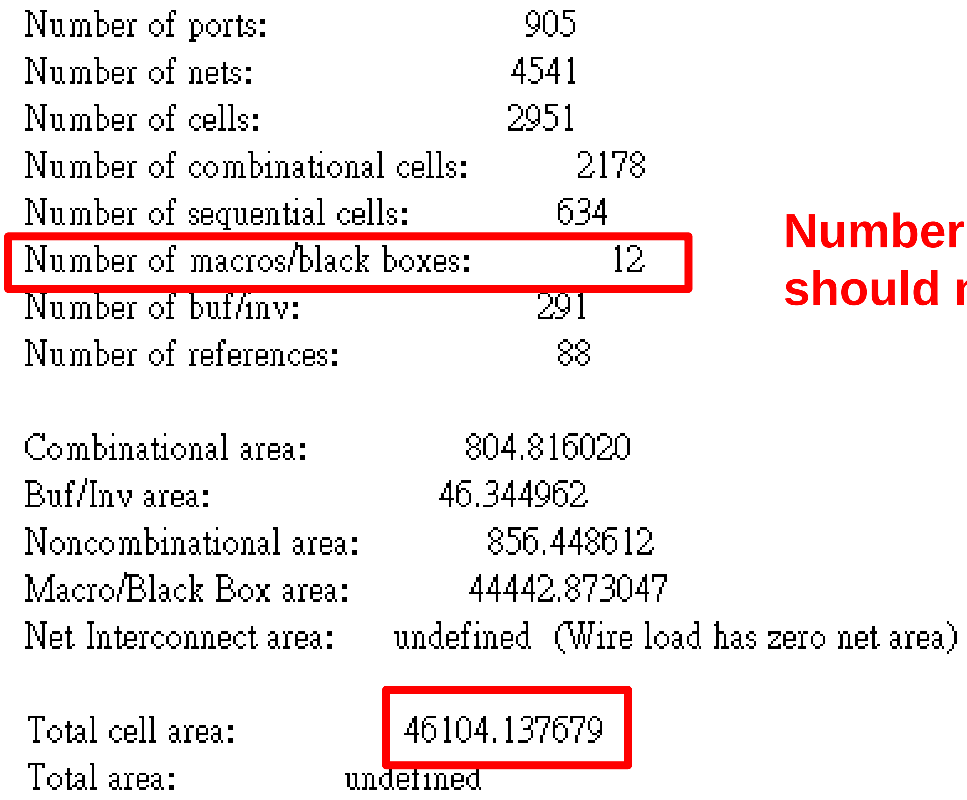
Time

- Time: Processing time from simulation (ex. 12583.50ns below)



```
../00_TESTBED/p0/pat0.dat
----------------------------------------------
-                    START                   -
----------------------------------------------
----------------------------------------------
-                  ALL PASS!                 -
----------------------------------------------
$finish called from file "../00_TESTBED/testbench.v", line 221.
$finish at simulation time             12583500
           V C S   S i m u l a t i o n   R e p o r t
Time: 12583500 ps
```

Area

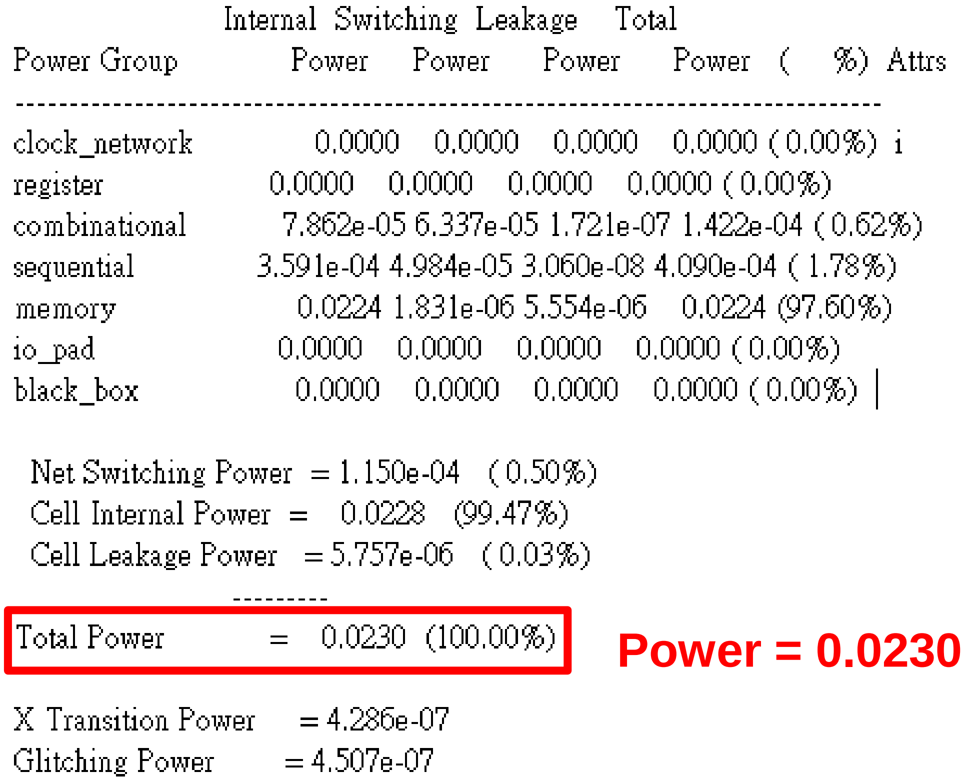


```
Number of ports:                    905
Number of nets:                    4541
Number of cells:                   2951
Number of combinational cells:         2178
Number of sequential cells:          634
Number of macros/black boxes:            12
Number of buf/inv:                  291
Number of references:                88

Combinational area:            804.816020
Buf/Inv area:               46.344962
Noncombinational area:           856.448612
Macro/Black Box area:         44442.873047
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:            46104.137679
Total area:             undefined
```

Number of macros/black boxes should not be 0

Power

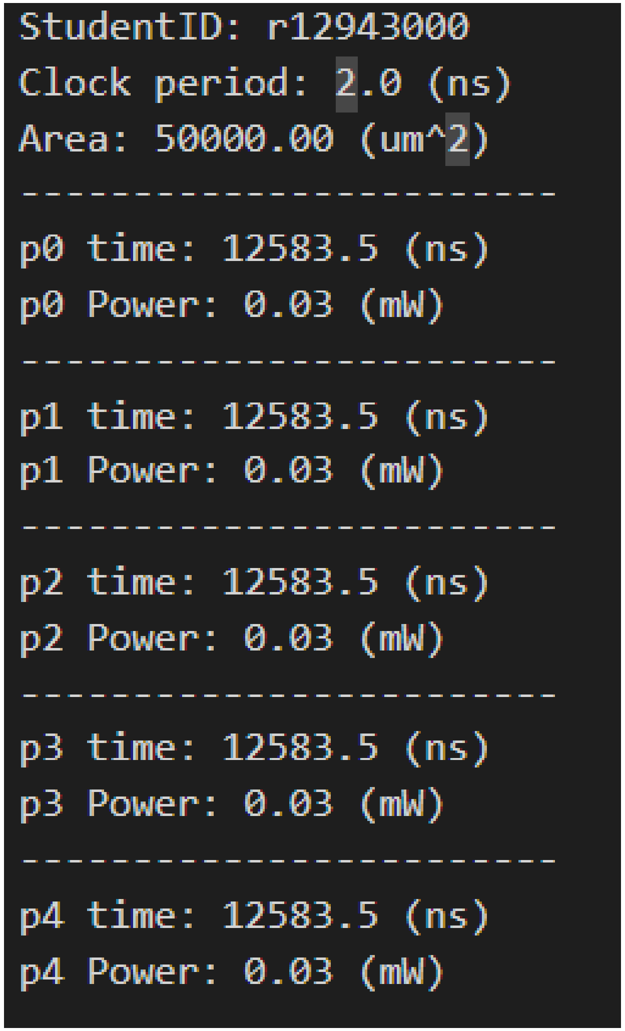
- Use Primetime to calculate power



```
                        Internal  Switching  Leakage    Total
Power Group               Power     Power     Power     Power  (   %)  Attrs
--------------------------------------------------------------------------
clock_network           0.0000    0.0000    0.0000    0.0000 ( 0.00%)  i
register             0.0000    0.0000    0.0000    0.0000 ( 0.00%)
combinational         7.862e-05 6.337e-05 1.721e-07 1.422e-04 ( 0.62%)
sequential          3.591e-04 4.984e-05 3.060e-08 4.090e-04 ( 1.78%)
memory                 0.0224 1.831e-06 5.554e-06    0.0224 (97.60%)
io_pad               0.0000    0.0000    0.0000    0.0000 ( 0.00%)
black_box              0.0000    0.0000    0.0000    0.0000 ( 0.00%)

  Net Switching Power  = 1.150e-04   ( 0.50%)
  Cell Internal Power  =    0.0228   (99.47%)
  Cell Leakage Power   = 5.757e-06   ( 0.03%)
                        ---------
Total Power            =    0.0230  (100.00%)

X Transition Power     = 4.286e-07
Glitching Power        = 4.507e-07
```

Power = 0.0230 mW

Report



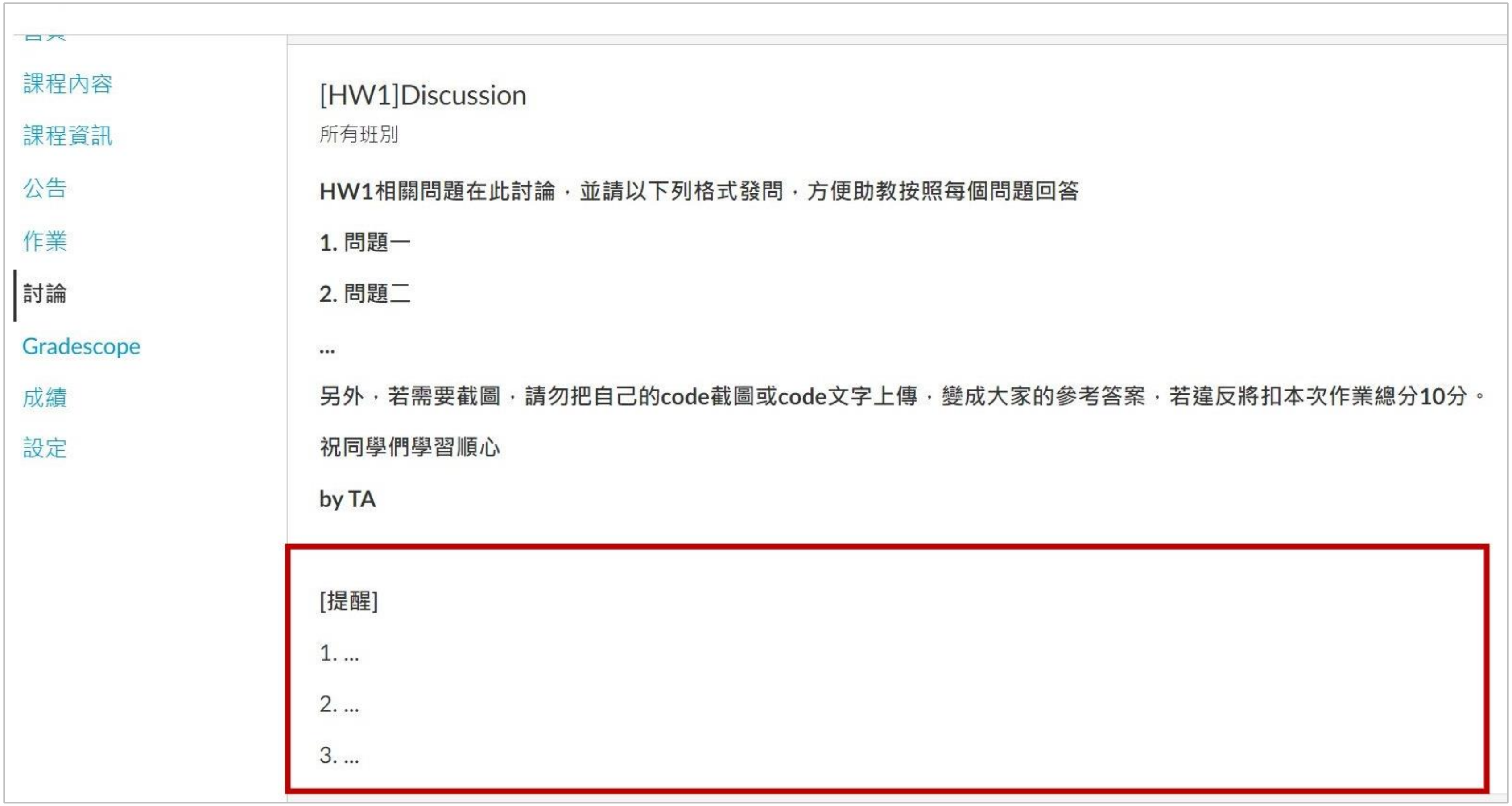
```
StudentID: r12943000
Clock period: 2.0 (ns)
Area: 50000.00 (um^2)
------------------------
p0 time: 12583.5 (ns)
p0 Power: 0.03 (mW)
------------------------
p1 time: 12583.5 (ns)
p1 Power: 0.03 (mW)
------------------------
p2 time: 12583.5 (ns)
p2 Power: 0.03 (mW)
------------------------
p3 time: 12583.5 (ns)
p3 Power: 0.03 (mW)
------------------------
p4 time: 12583.5 (ns)
p4 Power: 0.03 (mW)
```

Discussion

- **NTU Cool Discussion Forum**
  - For any questions not related to assignment answers or privacy concerns, please use the NTU Cool discussion forum.
  - **TAs will prioritize answering questions on the NTU Cool discussion forum**

- **Email:** r12943125@ntu.edu.tw
  - Title should start with **[ACVSD 2025 Spring HW1]**
  - Email with wrong title will be moved to trash automatically

Discussion



[HW1]Discussion

所有班別

HW1相關問題在此討論，並請以下列格式發問，方便助教按照每個問題回答

1. 問題一

2. 問題二

...

另外，若需要截圖，請勿把自己的code截圖或code文字上傳，變成大家的參考答案，若違反將扣本次作業總分10分。

祝同學們學習順心

by TA

[提醒]

1. ...

2. ...

3. ...

References

- [1] Reference for run-length encoding
  - Run-length Encoding
- [2] Reference for UPF
  - Synopsys ® Multivoltage Flow User Guide