EE213 Final Project

Guideline:

1) Please prepare your **submission in English**. No Chinese submissions will be accepted.

2) Please put all your **simulation results** as well as necessary **descriptions** in your report (.PDF) and submit it to Blackboard. Also, please pack **all the source files** to a zip with name 'Project_ID1_Name1_ID2_Name2', and submit the source file package to this link: (http://pan.shanghaitech.edu.cn/cloudservice/outerLink/decode?c3Vnb24xNjY4MzkxMDY4MzIzc3Vnb24=)

3) Please submit on time. Late submission will **NOT** be accepted.

4) Please **work on your own**. Discussion is permissible, but identical submissions are unacceptable! Here is the Academic Integrity policy of SIST. (https://sist.shanghaitech.edu.cn/2019/1107/c2842a46321/page.htm)

Timeline:

Dec. 11th: DDL of milestone report, complete schematic and pre-simulation parts
Dec. 18th: Check point for layout synthesis, submit your project for competition
Dec. 30th: DDL of final submission, complete the layout and post-simulation parts (passing DRC and LVS rule checks)

Design Task:

In this project, you will design a simple CPU to achieve a Max_pooling and a 2x2 matrix multiplication. The CPU mainly consists of 2 parts: 1) a comparator to perform Max_pooling operation, an adder and a multiplier for 2x2 matrix multiplication (all wrapped in one ALU module); 2) 20x10 bits SRAM for data storage. Each element of input matrices has 4-bit length while each element of output matrices has 10-bit length. The control unit should be designed by yourself. All numbers are **unsigned positive integers**. The final grade will be given based on: 1) your completion rate of the processing unit and 2) performance ranking.

You should use **45nm FreePDK** to design your module, draw the layout with **Cadence Virtuoso** and simulate your design in **HSPICE** simulator.

A Simple CPU with SRAM Cache

In this project, you need to design a simple CPU circuit with simplified pipeline that performs the following operations: **writing input data into memory, reading data from memory, conducting operations like comparison, addition and multiplication**. The input/output pins and their functions are specified below.

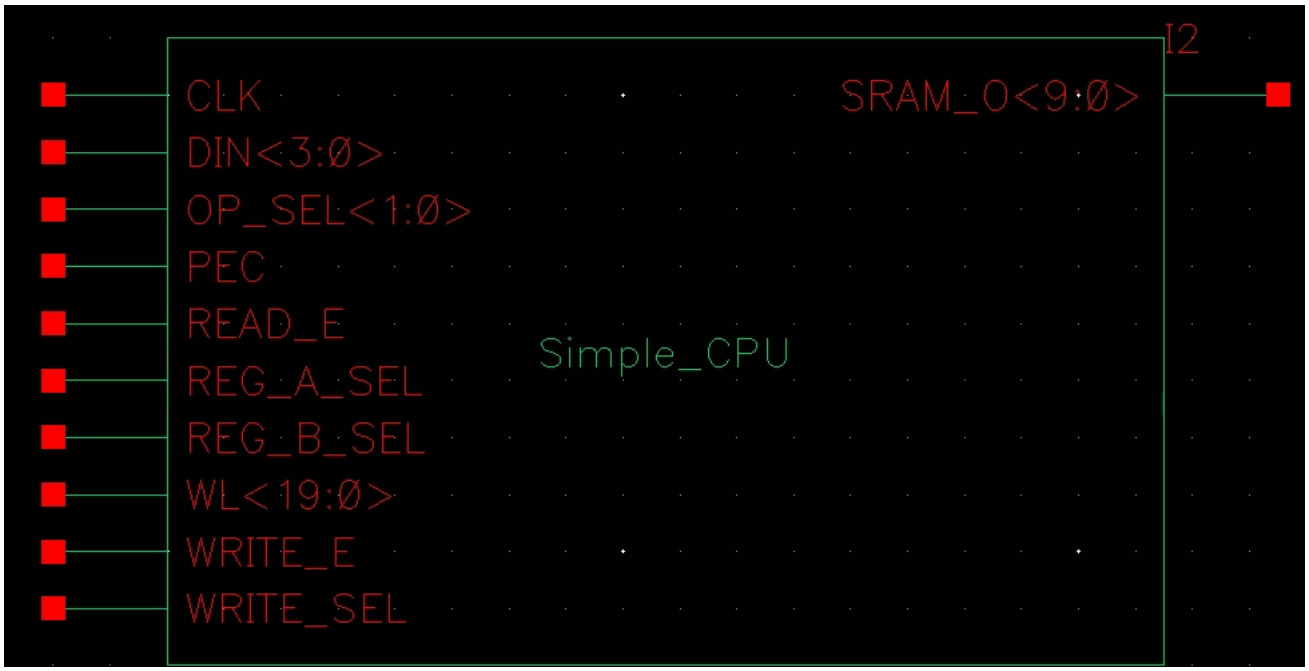

Figure 1 Symbol View Example of the Simple CPU

Input/Output

The following pins are **mandatory**.

NAME	FUNCTION	REMARKS
CLK	Clock signal	
DIN<3:0>	Data **input** signals	
READ_E	Enable SRAM data read	
WL<19:0>	Choose which SRAM row to read/write (Word Line)	There are 20 rows in total, the row decoder is omitted.
WRITE_E	Enable SRAM data write	
PEC	SRAM Cell precharge signal	
SRAM_O<9:0>	Data **output** from SRAM	Data are read from sense amplifiers.
OP_SEL<1:0>	Decide which operation to perform	11 for ADD, 10 for MUL, 01 for comparator

The following pins are **optional**, you can delete them or **add other control signals** based on your design.

NAME	FUNCTION	REMARKS
REG_IN1_SEL/REG_IN2_SEL	Decide whether to update data from SRAM (See Figure 2)	
WRITE_SEL	Decide which data to write into SRAM (DIN or ALU output)	

Computation Flow

You need to design a testbench that computes the following matrix multiplication:

$$\text{Max\_pooling}\begin{Bmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & A_{13} \\ A_{20} & A_{21} & A_{22} & A_{23} \\ A_{30} & A_{31} & A_{32} & A_{33} \end{Bmatrix} = \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix}, with\ stride = 2$$

$$\begin{bmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{bmatrix} \cdot \begin{bmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{bmatrix} = \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11} \end{bmatrix}$$

The example operation flow is:

1) Store A_{00} to C_{11} into SRAM.
2) Compute $B_{00} = \max\{\begin{matrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{matrix}\}$ and store B_{00} into SRAM.
3) Compute B_{01}, B_{10}, B_{11} and store them into SRAM.
4) Compute B_{00} * C_{00} and store the temporary result into SRAM.
5) Compute B_{01} * C_{10} and store the temporary result into SRAM.
6) Compute B_{00} * C_{00} + B_{01} * C_{10} and store D_{00} into SRAM.
7) Repeat 4) – 6) until all four results D_{00} - D_{11} are calculated.
8) Output D_{00} to D_{11} respectively.

You can customize the computation flow 2) - 7) to reduce the required cycles, but keep the input and output phase fixed. You may find Pattern Source Function (In HSPICE) useful in defining input signal pattern.

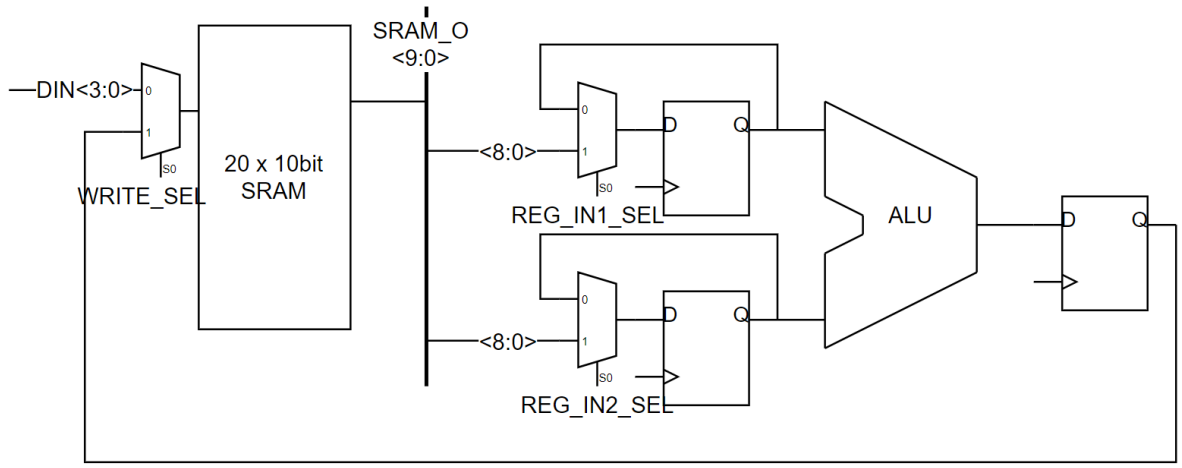

Figure 2 An Example Schematic of Simple CPU

ALU consists of a comparator, an adder and a multiplier.

Input Signal Setting

1) **Vdd = 0.8V**.
2) All data and control signals are generated by voltage sources. For your convenience, you can use **pattern source.**
3) The required frequency is **200MHz**, which means the **CLK** period is **5ns**.
4) The rise/fall time of all input signals including **CLK** is **100ps**.
5) You can arbitrarily set control signals.

Load Conditions

1) Unit sized inverter is Wp = 180nm, Wn = 90nm, Lp = Ln = 50nm.
2) Each bit of the outputs is loaded with CL = 0.1pF capacitance.

Gradings

1) Schematic and Pre-simulation: (40pts) (**Milestone report DDL: Dec. 11th, Sunday week 14**)
   a) Design schematics of the modules.
   b) Test the functionality of your modules by SPICE simulators.
2) Layout and Post-simulation: (40pts) (**Final report and source files DDL: Dec. 30th, Friday week 17**)
   a) Design/Modify schematics of the module.
   b) Draw the layout of your module and extract the netlist to simulate in SPICE simulators. Calculate the cycles, power and area and try to optimize them.
3) Competition (20pts)

In this part, we start the competition on **Dec. 18th (Sunday week 15)**. We will collect and show the completed teams' information of power, and cycle number (hide their personal information) on **Dec. 18th, Dec. 25th**, **Dec. 30th**.
The final grade of this part is based on the ranking. Therefore, all the teams will have a target to improve their performance.

- The area is defined as the area of the smallest external rectangle of your layout. **You should keep the area under 4000 um^2.**
- The cycle number is defined by the cycles needed from the first data input to the last output data read finished from SRAM.
- The power is defined as the average power of entire computation flow.

Timeline for Report Submission

1) Schematic and Pre-simulation:
   For the first stage, you should complete the schematic of CPU (all modules should be completed) and conduct corresponding pre-simulation. This report's DDL would be **Dec. 11th, Sunday week 14**. Your report should:
   - Explain in detail how you design.
   - Include schematic of your module in Virtuoso.
   - Verify your modules' functionality. (Make sure you simulate and test enough cases)

   Please submit your milestone report via BB. (Only one report is required for each group)
2) Competition:
   For competition, you can submit your complete project online since **Dec. 18th**, meaning that you have completed the layout and post-simulation parts in previous. The requirements for this submission and details for competition are provided below. This is not the final submission!
3) Layout and Post-simulation:
   For the final stage, you should complete the full project: schematic, layout that passes LVS and DRC, hspice files for pre-simulation and the corresponding screenshots, hspice files for post-simulation and the corresponding screenshots. The final report and source files' DDL would be **Dec. 30th, Friday week 17**. The final report should be serious and formal, with a clear format, including enough content and details. Your final report should include:
   - Details of your schematic design.
   - Layout (with the area information) of your whole module.
   - LVS, DRC and PEX screenshots showing no violations.
   - Plots of input and output waveforms of different input test cases.
   - The overall power, cycle number and area.

   Classify and pack all your source files into a compressed file and name it with the title: 'Project_ID1_Name1_ID2_Name2'.

Please submit your final report via BB and your source file package via this link:
(http://pan.shanghaitech.edu.cn/cloudservice/outerLink/decode?c3Vnb24xNjY4MzkxMDY4MzIzc3Vnb24=)

Competition

1) Test case:
   A: $\begin{bmatrix} 0111 & 0011 & 0010 & 1101 \\ 0101 & 0000 & 0100 & 0101 \\ 0110 & 0001 & 1010 & 1000 \\ 1001 & 0100 & 0111 & 1111 \end{bmatrix}$ C: $\begin{bmatrix} 0110 & 0111 \\ 1010 & 1111 \end{bmatrix}$ OUT: $\begin{bmatrix} 0010101100 & 0011110100 \\ 0011001100 & 0100100000 \end{bmatrix}$
   You should refine the voltage sources in your HSPICE files to simulate the above case. You need to set the simulation time to make sure all input, compute and output operations are finished.
2) Submit:
   a) Complete HSPICE files including testbench and functional circuits.
   b) Screenshots of total functional circuits layout. In the layout, you need to use the ruler to measure the minimum bounding **rectangle area**.
   c) Complete lab files.
   d) Make a file (such as readme.txt) to record the performance. You should record **area, power and cycle**

number.

3) To measure the **average power** of all circuits, use the following command:
   .meas tran pwr avg power
4) Compact all files and submit it to
   http://pan.shanghaitech.edu.cn/cloudservice/outerLink/decode?c3Vnb24xNjY4MzkxMTA1MTA0c3Vnb24=
5) We will start the first competition on Dec 18th. We will notify every team their results and rank. The overall results will be informed anonymously.