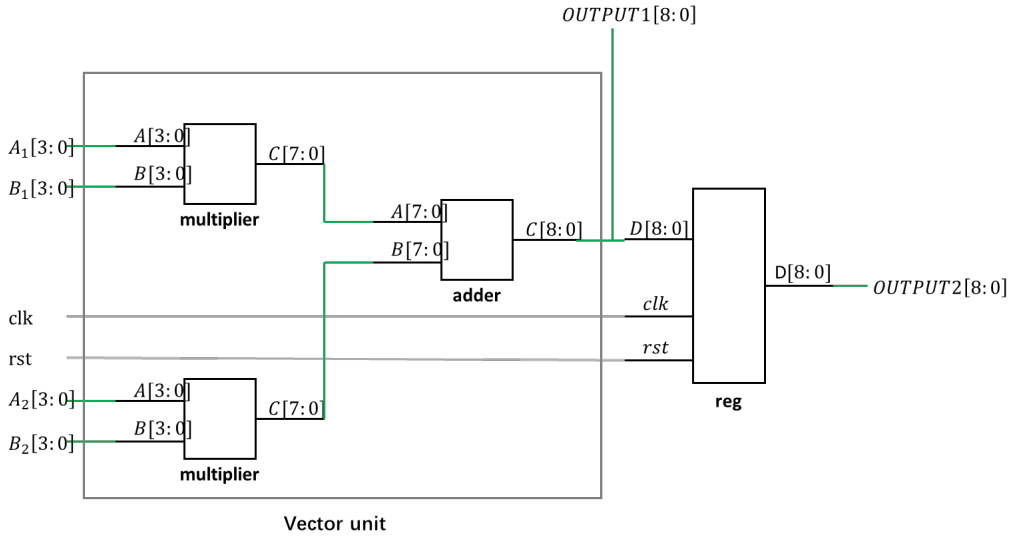
EE113 Final project - Matrix multiplication calculator

Basic requirement-Vector Unit

Introduction

In this part you need realize a simple Vector Unit designed for calculating matrix multiplication. This Vector Unit contains two 4-bit multipliers and an 8-bit adder. A register is appended to construct sequential logic.



Pin definition

NAME	FUNCTION
A_1	4-bit input.
B_1	4-bit input.
A_2	4-bit input.
B_2	4-bit input.
$OUTPUT1$	9-bit output. Combinatorial logic. MSB for carry (only for check).
$OUTPUT2$	9-bit output. Sequential logic. MSB for carry.
clk	Clock signal.
rst	Reset signal. Synchronous reset. High level trigger.

Matrix multiplication operation

$$\begin{bmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{bmatrix} \begin{bmatrix} n_{00} & n_{01} \\ n_{10} & n_{11} \end{bmatrix} = \begin{bmatrix} l_{00} & l_{01} \\ l_{10} & l_{11} \end{bmatrix}$$

$$\begin{aligned} l_{00} &= m_{00}n_{00} + m_{01}n_{10} \quad opt\_1 \\ l_{01} &= m_{00}n_{01} + m_{01}n_{11} \quad opt\_2 \\ l_{10} &= m_{10}n_{00} + m_{11}n_{10} \quad opt\_3 \\ l_{11} &= m_{10}n_{01} + m_{11}n_{11} \quad opt\_4 \end{aligned}$$

Example

cycle	operation	A_1	B_1	A_2	B_2	$OUTPUT$
1	opt_1	0001	0001	0010	0011	X

2	opt_2	0001	0010	0010	0100	0 0000 0111
3	opt_3	0011	0001	0100	0011	0 0000 1010
4	opt_4	0011	0010	0100	0100	0 0000 1111
5						0 0001 0110

Requirement

- The circuit functions properly in pre-simulation(Minimum standard: tp_vu⩽3ns, tp_reg⩽1ns).
- Pass the DRC and LVS check.
- The circuit functions properly in post-simulation. (Minimum standard: tp_vu⩽6ns, tp_reg⩽2ns).

Reference reading

Array Multiplier in Digital Logic

Brief introduction

An array multiplier is a digital combinational circuit used for multiplying two binary numbers by employing an array of full adders and half adders. This array is used for the nearly simultaneous addition of the various product terms involved. To form the various product terms, an array of AND gates is used before the Adder array.

Checking the bits of the multiplier one at a time and forming partial products is a sequential operation that requires a sequence of add and shift micro-operations. The multiplication of two binary numbers can be done with one micro-operation by means of a combinational circuit that forms the product bits all at once. This is a fast way of multiplying two numbers since all it takes is the time for the signals to propagate through the gates that form the multiplication array. However, an array multiplier requires a large number of gates, and for this reason it was not economical until the development of integrated circuits.

Implementation of array multiplier

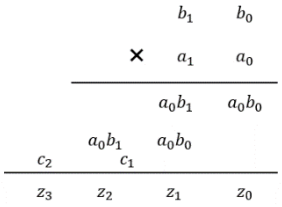
Consider the multiplication of two 2-bit numbers as shown in figure. The multiplicand bits are b_1 and b_0, the multiplier bits are a_1 and a_0, and the product is $z_3z_2z_1z_0$.

$z_0 = a_0 \cdot b_0$

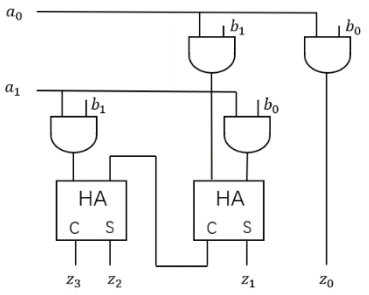
$z_1 = a_0 \cdot b_1 + a_1 \cdot b_0$

$z_2 = a_1 \cdot b_1 + c_1$

$z_3 = c_2$

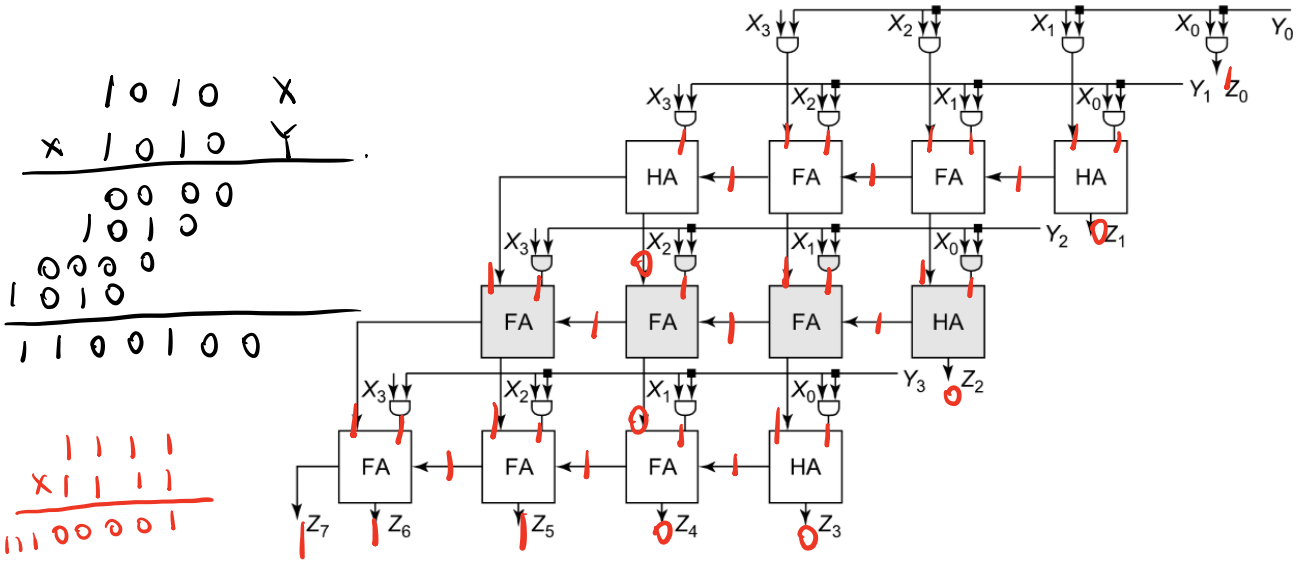


For the above multiplication, an array of four AND gates are required to form the various product terms like a_0b_0 etc. and then an adder array is required to calculate the sums involving the various product terms and carry combinations mentioned in the above equations to get the final Product bits.



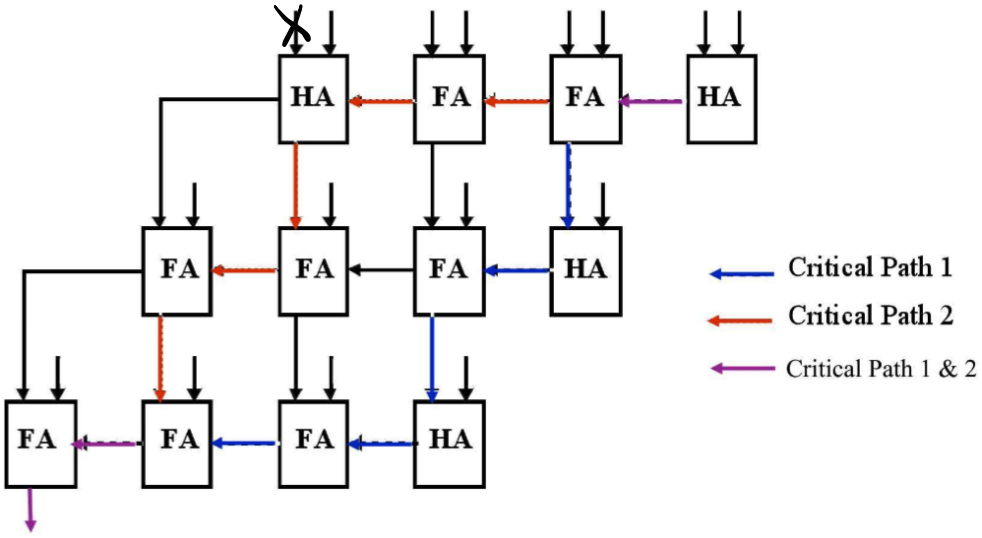
1. The first partial product is formed by multiplying a_0 by b_1, b_0. The multiplication of two bits such as a_0 and b_0 produces a 1 if both bits are 1; otherwise, it produces 0. This is identical to an AND operation and can be implemented with an AND gate.
2. The first partial product is formed by means of two AND gates.
3. The second partial product is formed by multiplying a_1 by b_1, b_0 and is shifted one position to the left.
4. The above two partial products are added with two Half-Adder (HA) circuits. Usually there are more bits in the partial products and it will be necessary to use full-adders to produce the sum.
5. Note that the least significant bit of the product does not have to go through an adder since it is formed by the output of the first AND gate.

A combinational circuit binary multiplier with more bits can be constructed in similar fashion. A bit of the multiplier is ANDed with each bit of the multiplicand in as many levels as there are bits in the multiplier. The binary output in each level of AND gates is added in parallel with the partial product of the previous level to form a new partial product. The last level produces the product.

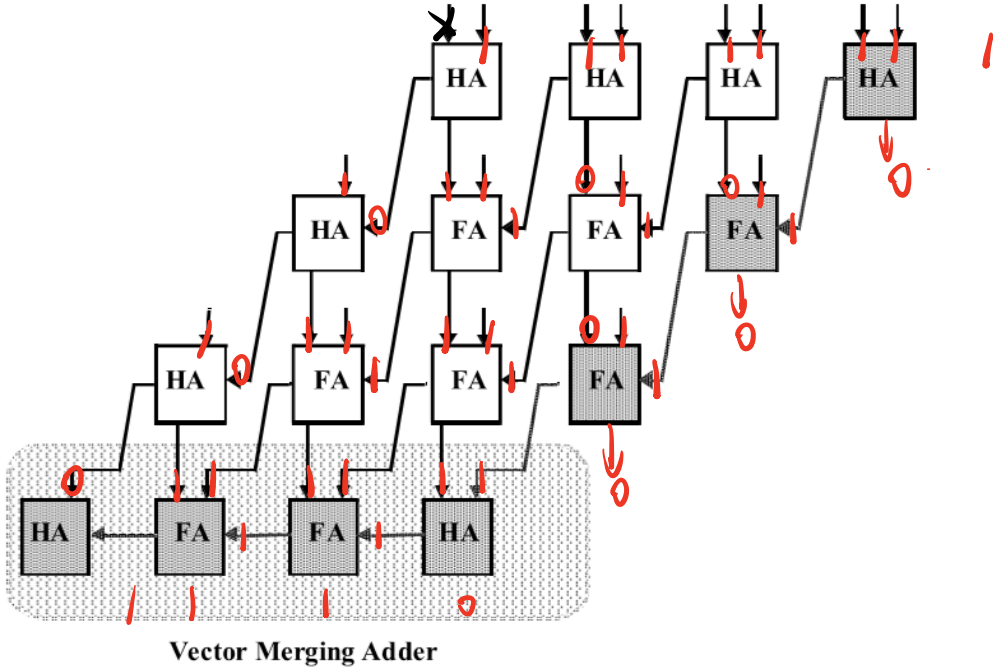


Further self-study: Carry save multiplier

A method to optimize $M \times N$ Array Multiplier. Analysis the critical path:



Array Multiplier: $t_{mult} \approx [(M-1)+(N-2)] \cdot t_{carry} + (N-1) \cdot t_{sum} + t_{and}$



Carry save multiplier: $t_{mult} \approx t_{and} + (N-1) \cdot t_{carry} + t_{merge}$

Bonus part-SRAM

Static Random-Access Memory (SRAM) is one of the important on-chip storage modules. In the bonus part, you are going to design and implement an 8×16 SRAM module in both the circuit and layout level and integrate it with the basic part you implemented before.

Introduction

Narrowly speaking, SRAM could be referred to its bitcell. In this project, the implementation of the bitcell should be the typical 6T structure, as shown in Fig. 1(a), compulsorily. Generally, apart from the core bitcell array, SRAM includes the row and column circuitry for reading and

writing operations.

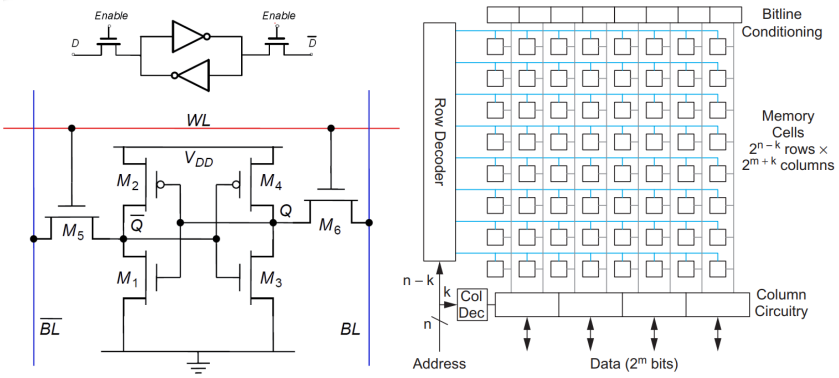


Fig. 1 (a) 6T CMOS SRAM cell and (b) SRAM bitcell array with peripheral circuitry.

The working principle of the SRAM cell is (or will be) discussed detailly in class. As for the peripheral circuitry, the row circuit may contain an address decoder and wordline buffers. The decoder takes an address as the input and outputs a one-hot wordline signal to activate a specific row of bitcells. A wordline buffer is needed to drive the large input capacitance of the row of cells (typically much more than sixteen in a real case).

The column module is usually consisted of the pre-charger, the write drivers, the sense amplifiers, and optional column multiplexers. The pre-charger (and the write drivers) properly conditions the bitlines before a read (write) operation. Sense amplifiers, a kind of analog circuit, are applied to each column of cells to accelerate the reading process. A reference structure for sense amplifier is given in Fig. 2, and you can freely use any other type that works. The optional multiplexers are used to select which columns of data to output.

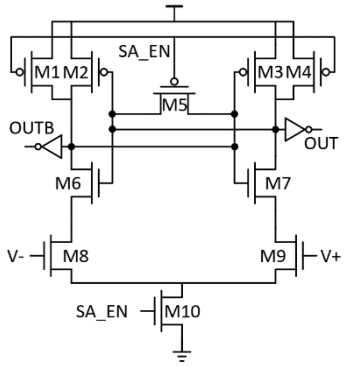


Fig. 2 A sense amplifier structure FYR.

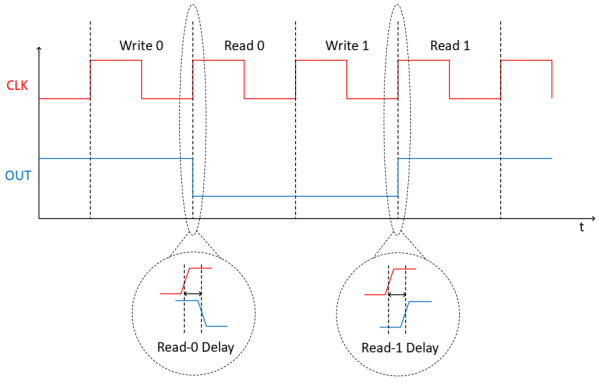


Fig. 3 The timing diagram of the read-after-write example.

Except for the I/O ports of the top module, the design and interconnection of the submodules is not required to be same. For more information on SRAM cell and the peripheral circuitry, you may refer to the lecture slides or other materials like the Section 12.2 of *CMOS VLSI Design: A Circuits and Systems Perspective(4th)* by Neil H. E. Weste and David Money Harris.

Bonus I

You should design and implement an 8×16 SRAM array with peripheral circuitry in schematic. A testbench is also needed.

Hint: You may need buffers and MUX to properly assign several enable signals in sequence in different operation modes.

Pins definition:

NAME	FUNCTION
Clk	Clock signal.
Data_in[15:0]	Data to write.
Data_out[15:0]	Read data.
mode	Write-or-read.

Requirement:

- The circuit functions properly in pre-simulation.
- Each width of the transistor in bitcells is no larger than 800nm.
- The clk signal should only act as control signal.
- The output should not change before another read operation even after a write (Fig. 3).
- For the column circuitry, during the read-mode, the voltage of the discharged BL/BLB is no lower than 70% VDD.

Bonus II

Implement the layout in Bonus I and the pins are as defined.

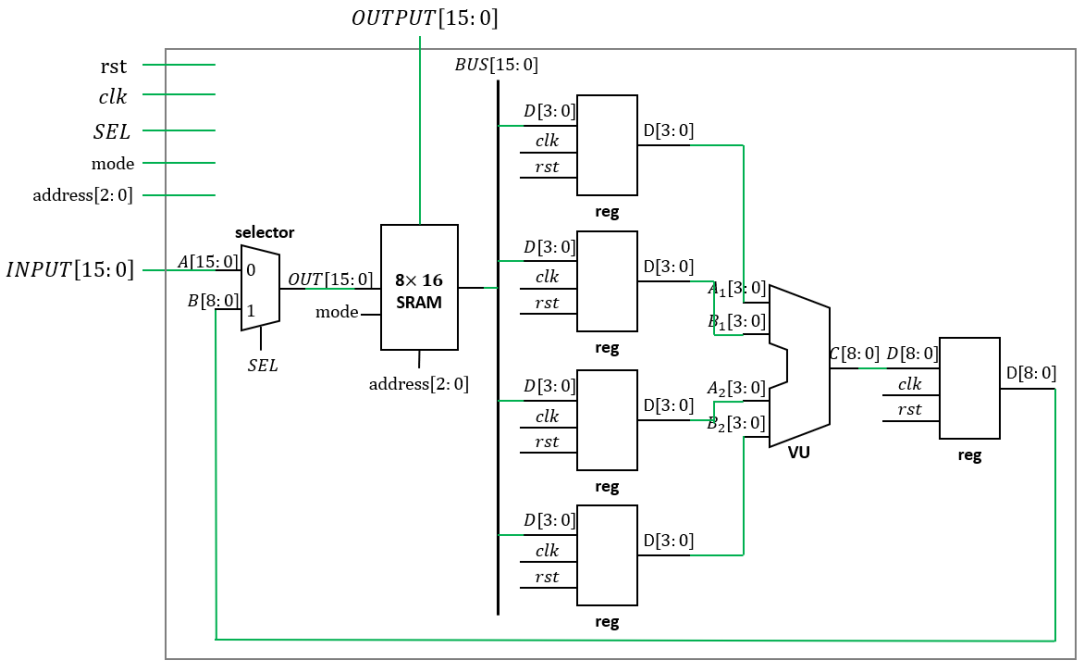
Requirement:

- Pass the DRC and LVS check.
- The circuit functions properly in post-simulation.
- One body tap at least in every eight bitcells in the horizontal or vertical direction.

BonusIII - Matrix multiplication calculator

Function

1. Write the INPUT data to SRAM.
2. Read INPUT data from SRAM to Vector Unit.
3. Write the OUPUT data back to SRAM.



Pin definition

NAME	FUNCTION
$INPUT$	16-bit input.
$OUTPUT$	9-bit output. MSB for carry.
clk	Clock signal.
rst	Reset signal. Synchronous reset. High level trigger.
SEL	Decide which data to write into SRAM
$mode$	Decide whether read or write operation to SRAM.
$address$	3-bit input. Address for SRAM.

INPUT example

$$\begin{bmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{bmatrix} \begin{bmatrix} n_{00} & n_{01} \\ n_{10} & n_{11} \end{bmatrix} = \begin{bmatrix} l_{00} & l_{01} \\ l_{10} & l_{11} \end{bmatrix}$$

SRAM storage example

$m_{00}\ \ n_{00}\ \ m_{01}\ \ n_{10}$
$m_{00}\ \ n_{01}\ \ m_{01}\ \ n_{11}$
$m_{10}\ \ n_{00}\ \ m_{11}\ \ n_{10}$
$m_{10}\ \ n_{01}\ \ m_{11}\ \ n_{11}$
l_{00}
l_{01}
l_{10}
l_{11}

Week Plan

The following plan is just for reference, you can assign the detail task by yourself, remember

to reserve enough time to draw layout.

Week1	Draw the schematic and the layout of 8-bit adder. Draw the schematic of register.
Week2	Learn the architecture of multiplier. Draw the schematic of multiplier. Draw the layout of register.
Week3	Draw the layout of multiplier. Finish basic part.
Week4	Learn SRAM design. Finish Bonus Ⅰ.
Week5	Finish Bonus Ⅱ.
Week6	Finish BonusⅢ.