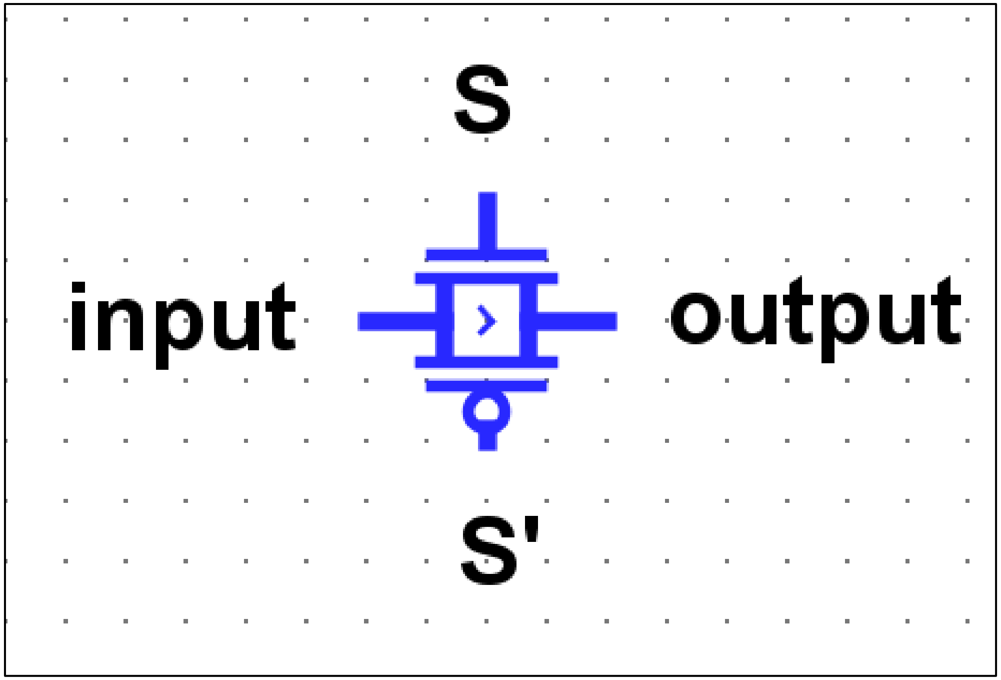
Computer Architecture Lab

Lab04 – Week #4

MPLAB
Media Processing Laboratory

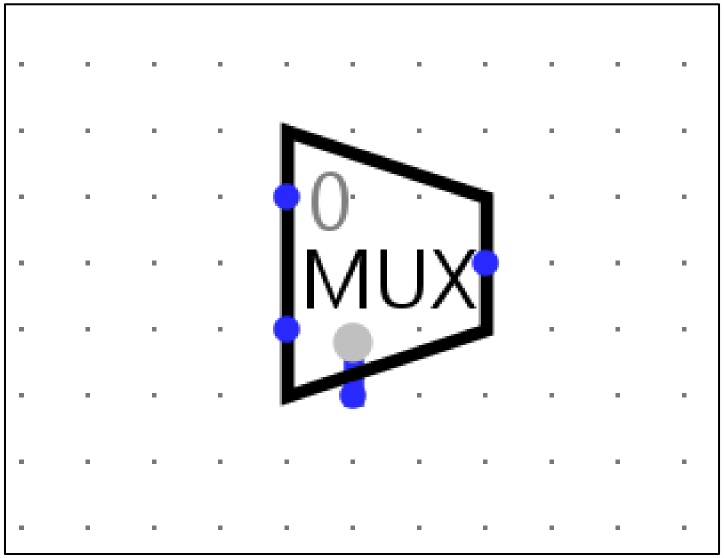
컴퓨터 공학과 이성원 교수님 연구실

MUX using Transmission Gate



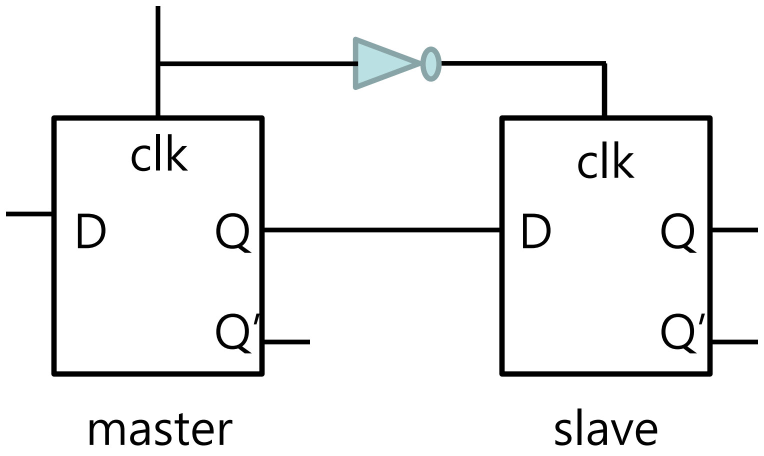
- ➢ You should consider Transmission gate's feature and MUX's function
- ➢ Input 1 goes output when S = 1
- ➢ Input 2 goes output when S`= 1

Latch using MUX



- You should consider The function of the latch storing the value
- You should consider the number of MUX's input
- Based on D latch, the number of input is two, D and enable
- It is important how to connect the remaining input to keep the values stored

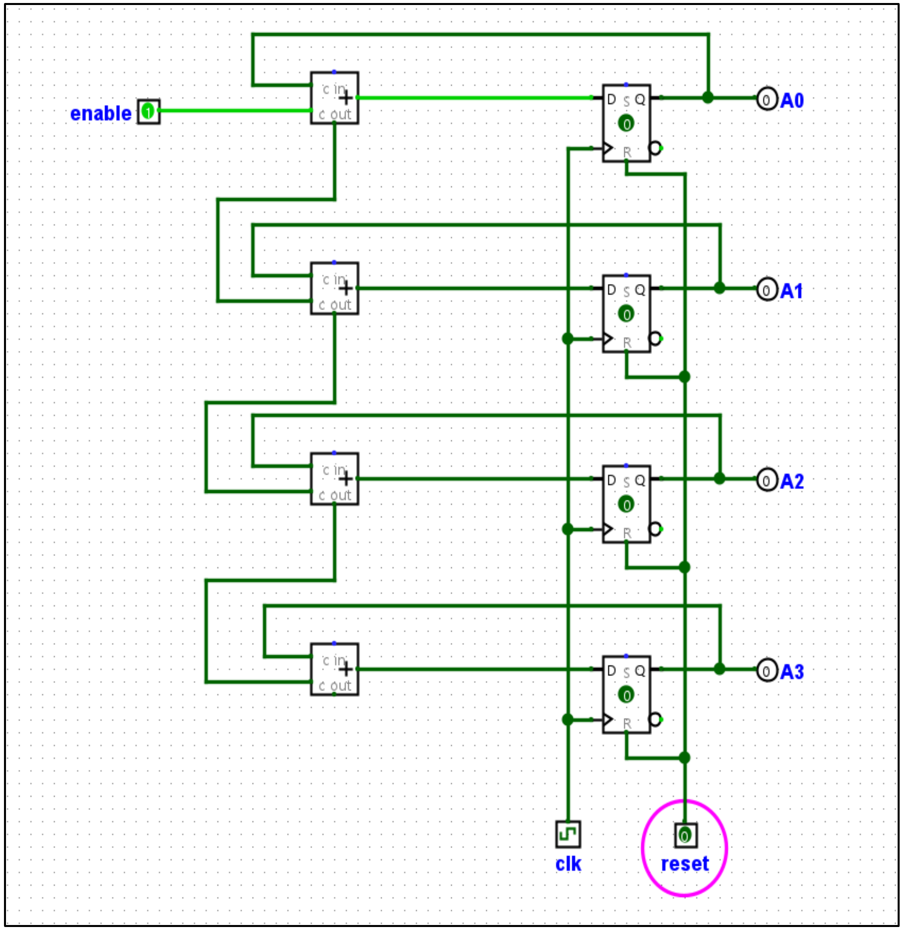
D-Flip Flop using Latch



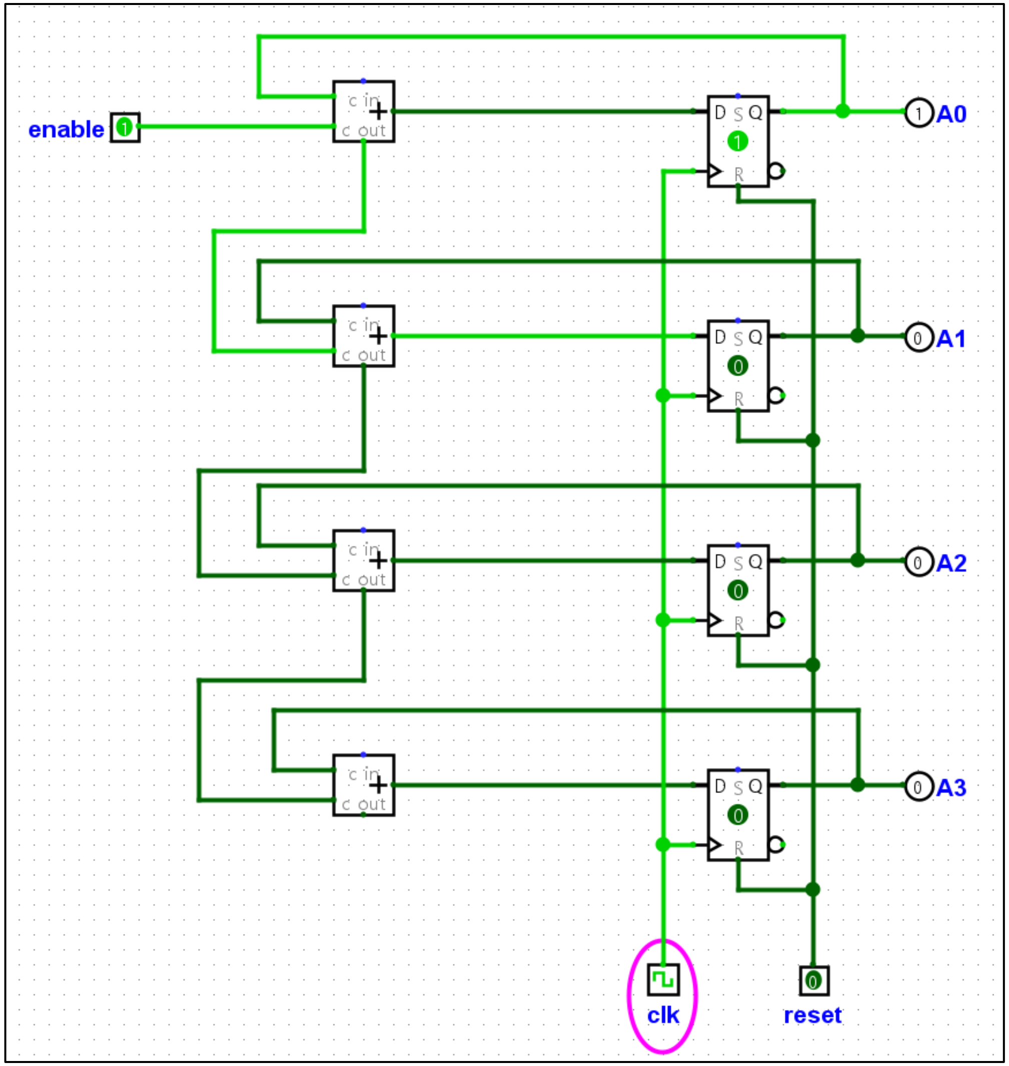
- It is easy to design considering the form of Master-slave D-Flip Flop
- FF design uses the method of Latch design with Mux
- When connecting the input, you should consider that master's output enters slave's input
- Master operates when clock is 1, and slave operates when clock is 0

Synchronous Up-Counter Using D-FF&Adder (1/5)

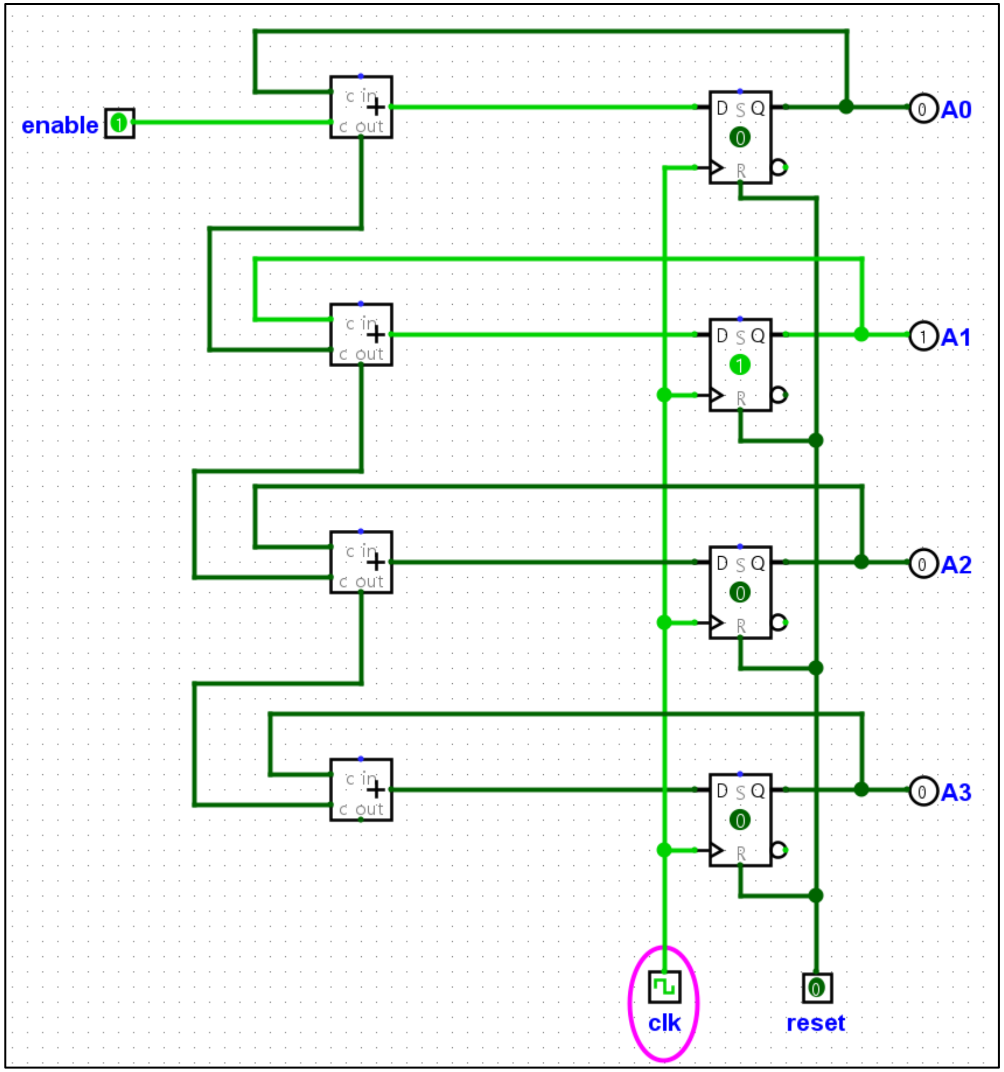
- You should consider pushing up the value through c_in and c_out of the Adder



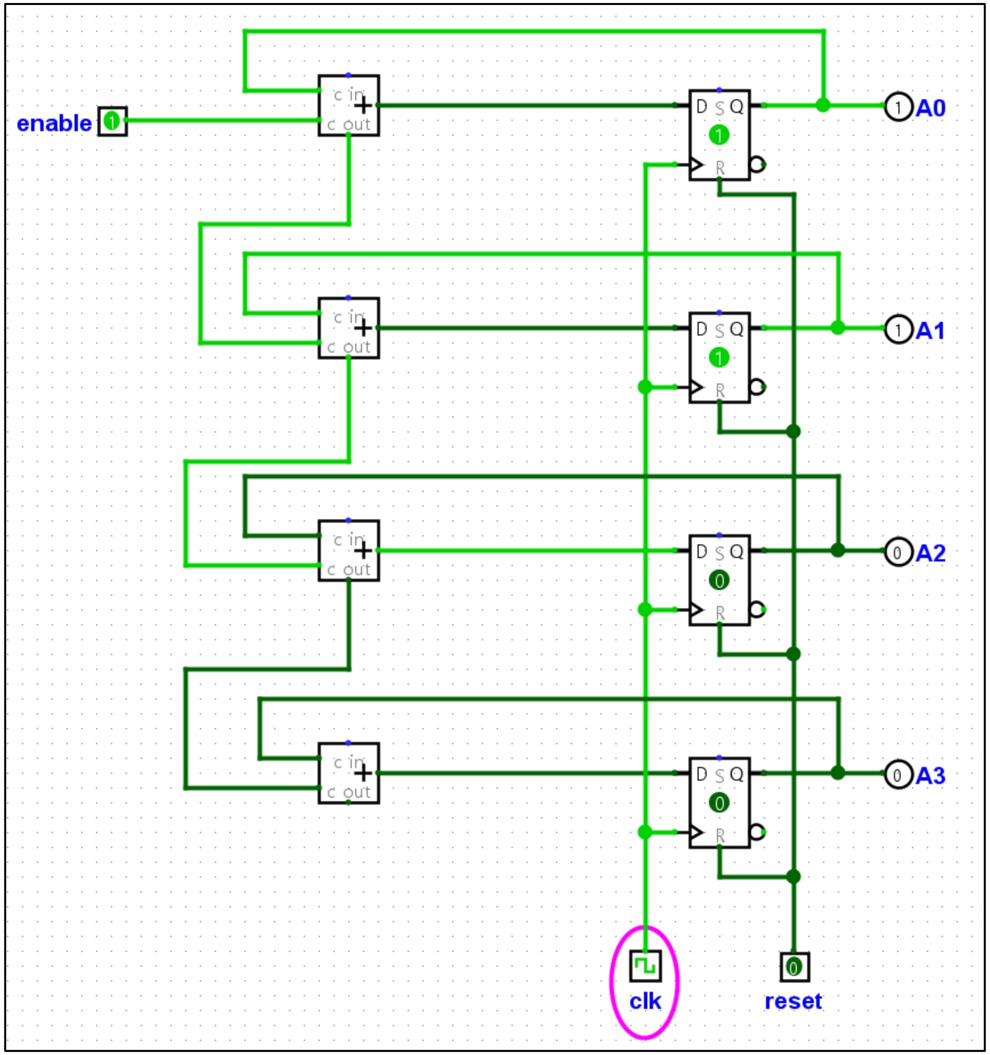
Synchronous Up-Counter Using D-FF&Adder (2/5)



Synchronous Up-Counter Using D-FF&Adder (3/5)



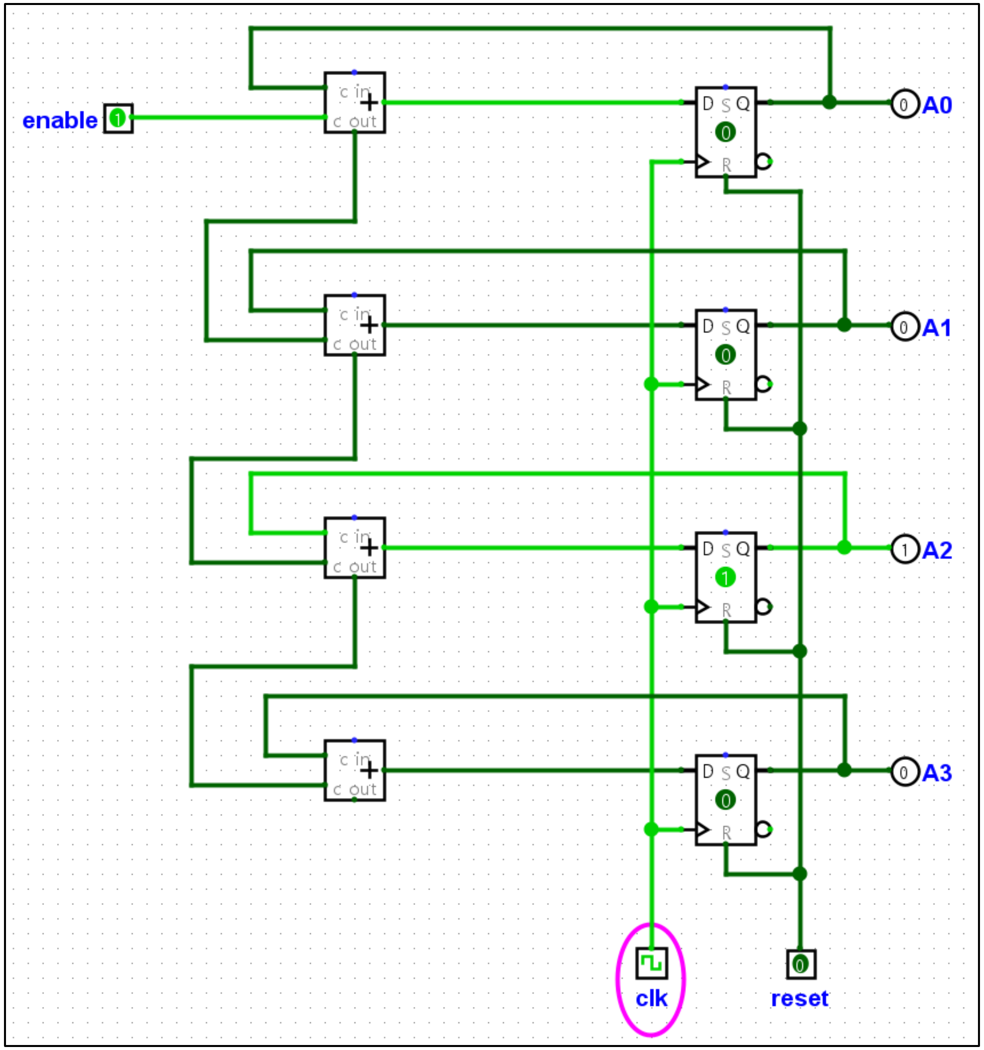
Synchronous Up-Counter Using D-FF&Adder (4/5)



Synchronous Up-Counter Using D-FF&Adder (5/5)

- When you have to design Synchronous up-down counter, recommend to use mux to control input value

- Point is to use c_in

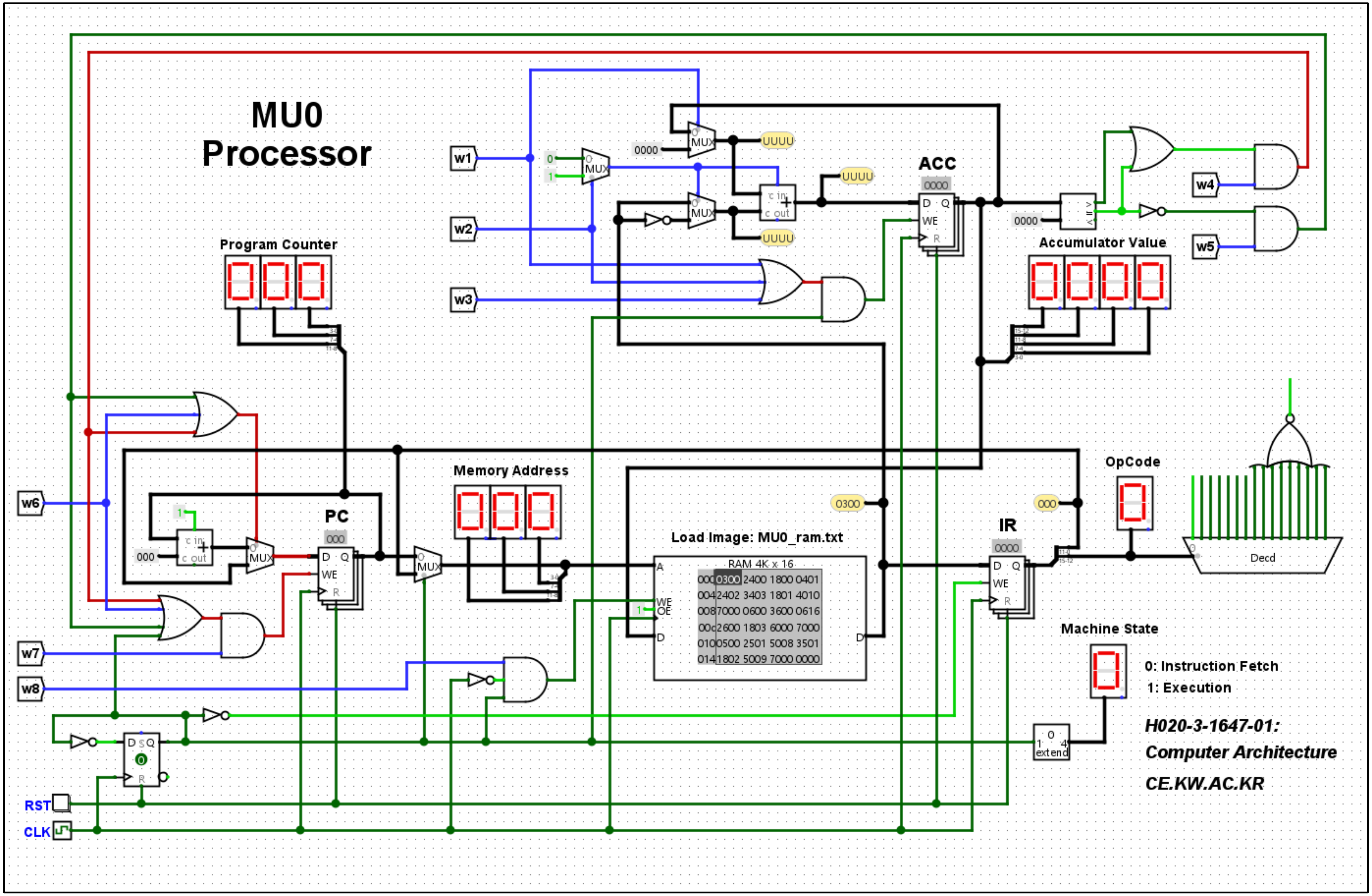


MPLAB
Media Processing Laboratory

MU0 Instructions

- op-code 0000 (LDA S): load into accumulator, operand from memory location addressed as ‘S’
- op-code 0001 (STO S): store the content of accumulator register at the memory location addressed as ‘S’
- op-code 0010 (ADD S): ACC = ACC + [S] add content of memory location addressed as ‘S’ with content of accumulator and save result at accumulator
- op-code 0011 (SUB S): ACC = ACC – [S]
- op-code 0100 (JMP S): jump to the memory location addressed as ‘S’. This will unconditionally change the program flow. PC = [S]
- op-code 0101 (JGE S): if ACC>=0, then jump.
  This is conditional jump with condition that ALU result is positive
- op-code 0110 (JNE S): another conditional jump with condition that ACC content is non-zero
- op-code 0111 (STP): stop the program execution

MU0 Processor



Load Image: MU0_ram.txt

```
RAM 4K x 16
000 0300 2400 1800 0401
004 2402 3403 1801 4010
008 7000 0600 3600 0616
00c 2600 1803 6000 7000
010 0500 2501 5008 3501
014 1802 5009 7000 0000
```

0: Instruction Fetch

1: Execution

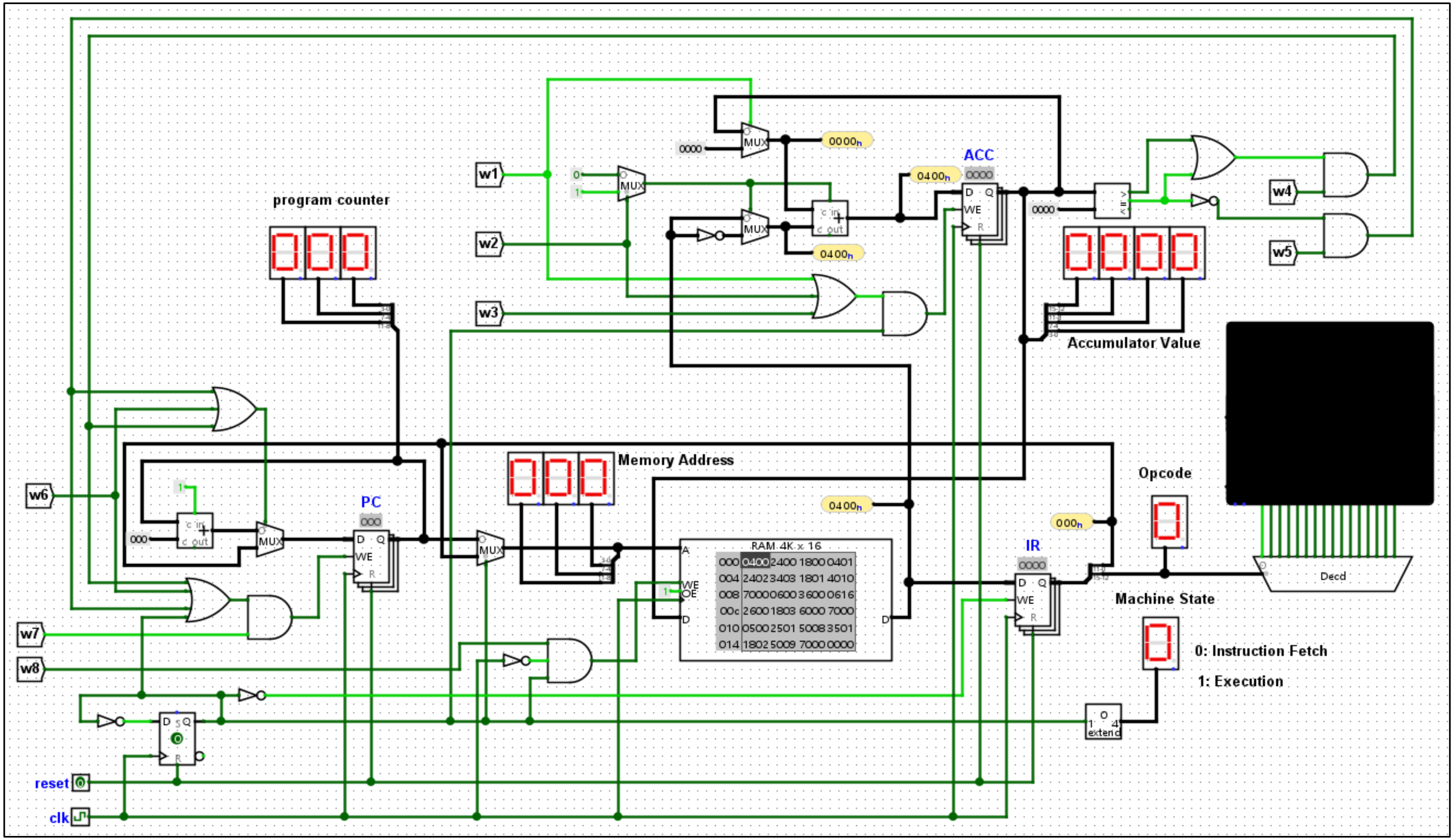
H020-3-1647-01:

Computer Architecture

CE.KW.AC.KR

Ex) Load (1/3)

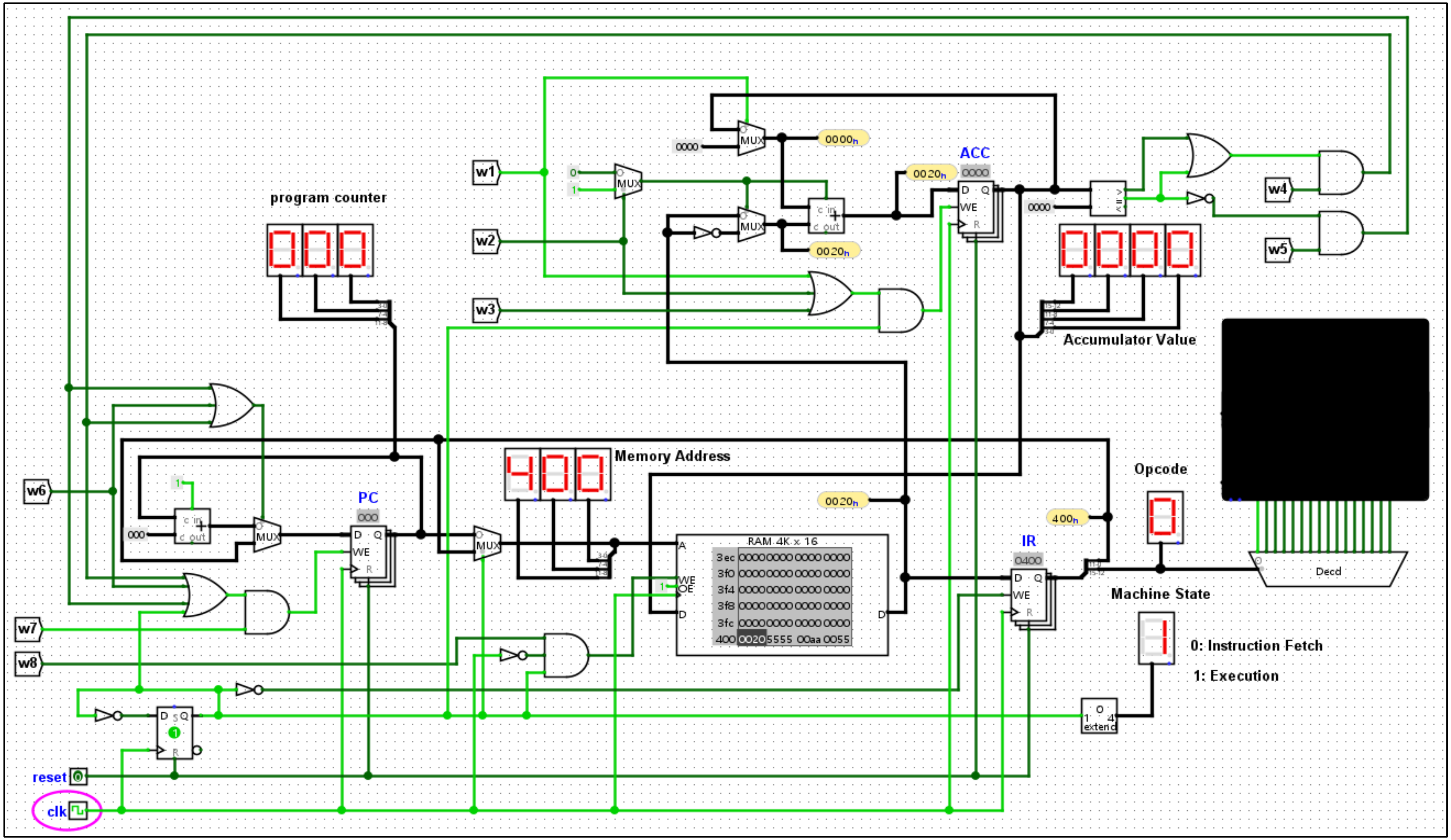
- Fetch



```
000 0400 2400 1800 0401
004 2402 3403 1801 4010
008 7000 0600 3600 0616
00c 2600 1803 6000 7000
010 0500 2501 5008 3501
014 1802 5009 7000 0000
```

Ex) Load (2/3)

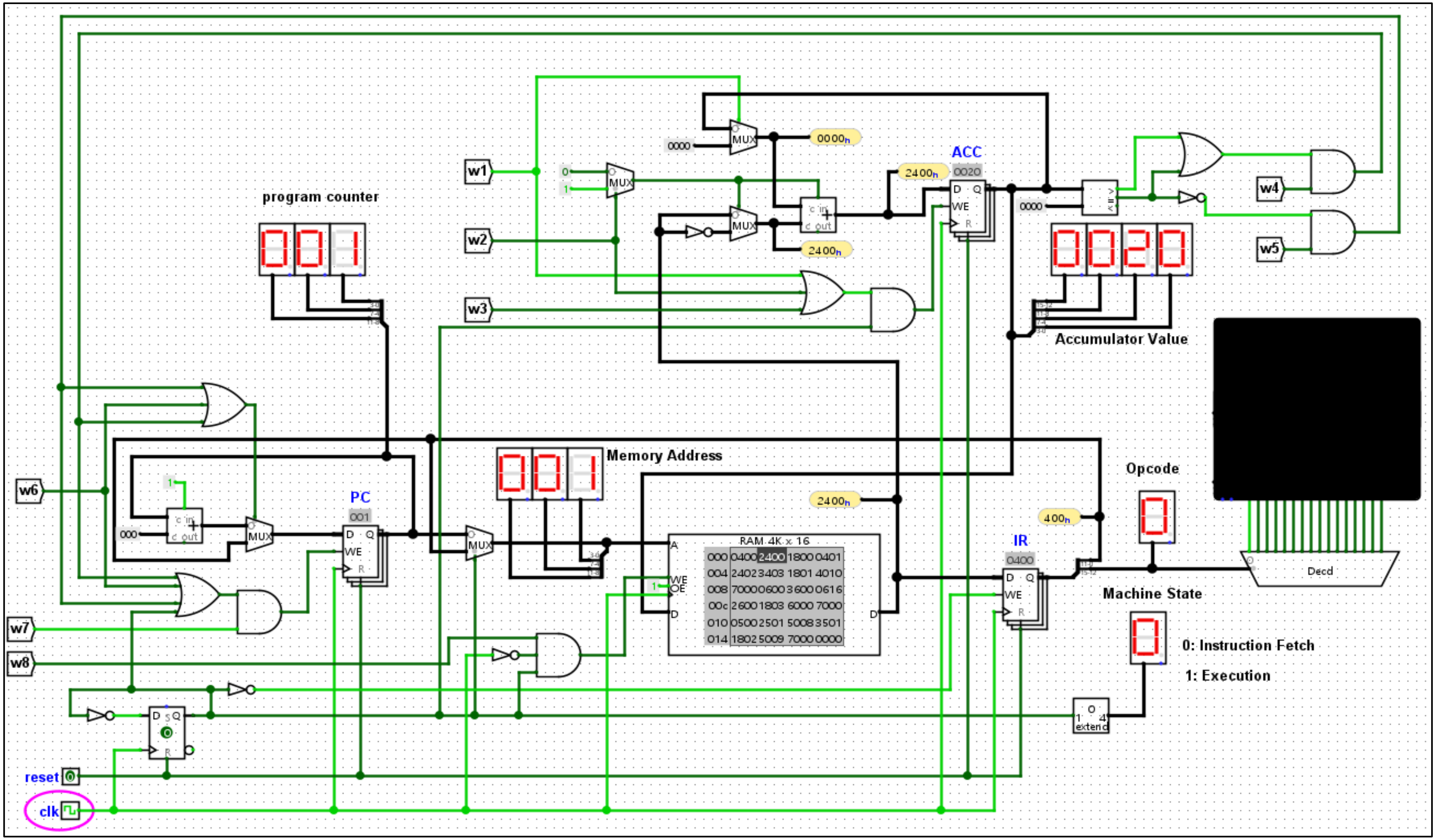
- Execution



```
RAM 4K x 16
3ec 0000 0000 0000 0000
3f0 0000 0000 0000 0000
3f4 0000 0000 0000 0000
3f8 0000 0000 0000 0000
3fc 0000 0000 0000 0000
400 0020 5555 00aa 0055
```

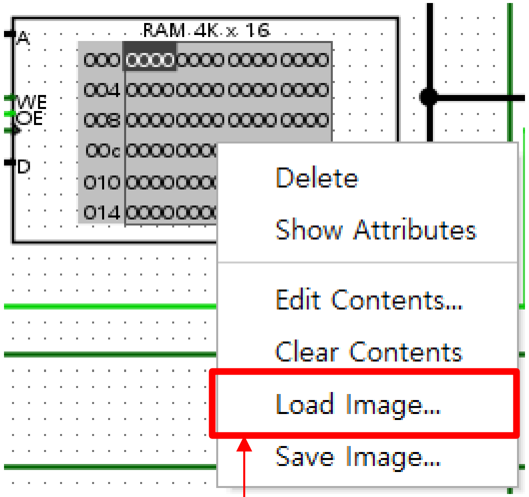
Ex) Load (3/3)

- Next Fetch



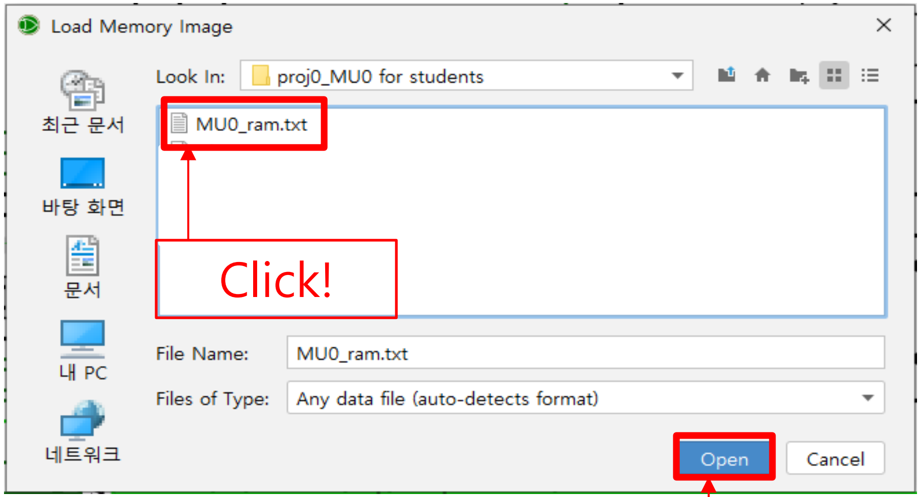
```
RAM 4K x 16
000 0400 2400 1800 0401
004 2402 3403 1801 4010
008 7000 0600 3600 0616
00c 2600 1803 6000 7000
010 0500 2501 5008 3501
014 1802 5009 7000 0000
```

How to use MU0.txt



```
000 0000 0000 0000 0000
004 0000 0000 0000 0000
008 0000 0000 0000 0000
00c 0000000
010 0000000
014 0000000
```

Click!



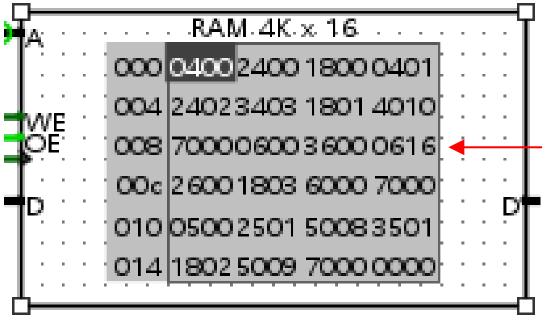
Look In: proj0_MU0 for students

File Name: MU0_ram.txt

Files of Type: Any data file (auto-detects format)

Click!

Click!



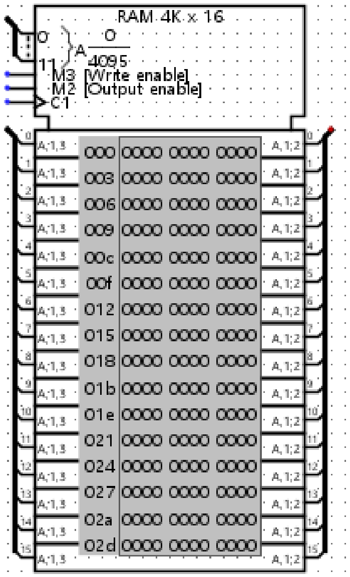
```
000 0400 2400 1800 0401
004 2402 3403 1801 4010
008 7000 0600 3600 0616
00c 2600 1803 6000 7000
010 0500 2501 5008 3501
014 1802 5009 7000 0000
```

Loaded

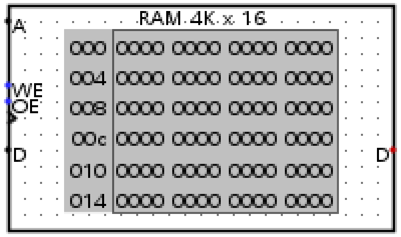
Point to note

- ➢ Be careful of ram settings
- ➢ Be careful of number of bits
- ➢ You can also change appearance

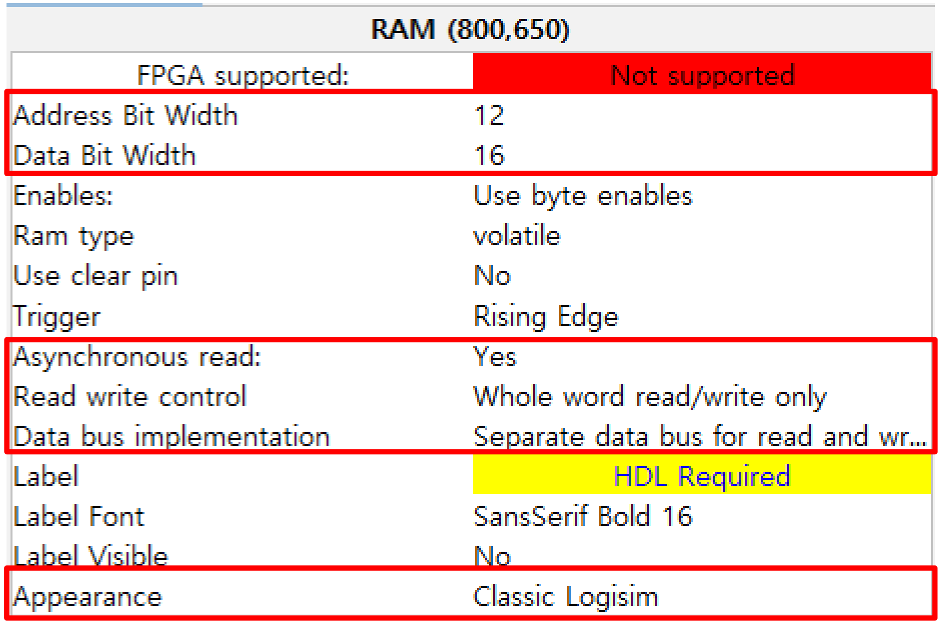
✓ Logisim-evolution



✓ Classic logisim



✓ RAM



RAM (800,650)	
FPGA supported:	Not supported
Address Bit Width	12
Data Bit Width	16
Enables:	Use byte enables
Ram type	volatile
Use clear pin	No
Trigger	Rising Edge
Asynchronous read:	Yes
Read write control	Whole word read/write only
Data bus implementation	Separate data bus for read and wr...
Label	HDL Required
Label Font	SansSerif Bold 16
Label Visible	No
Appearance	Classic Logisim

Thank You