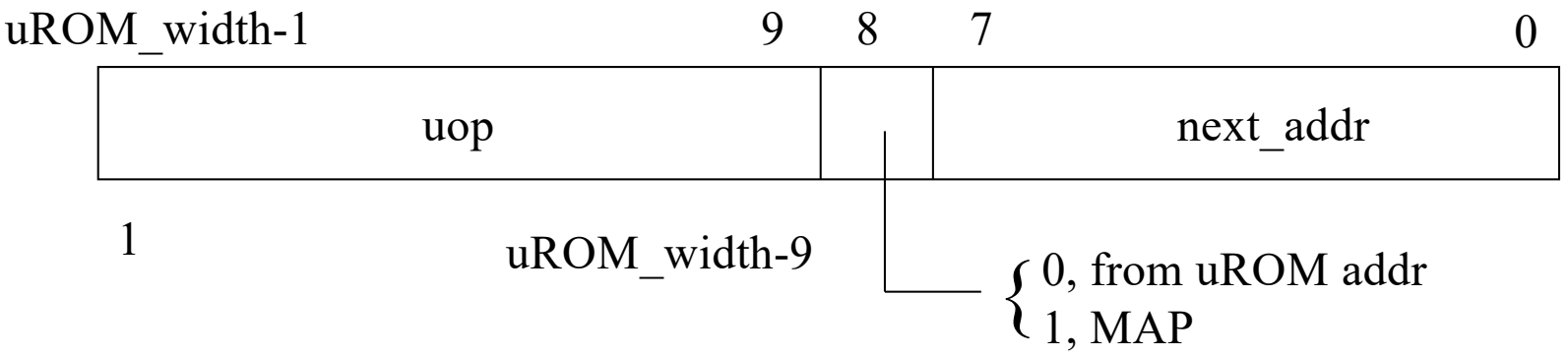
Important Dates

- 12/11 Last date to demo lab 7 and upload project code (Zip file of everything)
- 12/15 (Monday) 9:00am-11:00am, 106 KUPR, Final
- 12/17 (Wednesday, 11:59pm) Last day to submit report –**No late submission would be accepted**

Assignments

- Week 3
  - Develop the microcode for your control unit (submit as post-lab). Use the excel template (sample uROM)
  - Start working on the VHDL code.

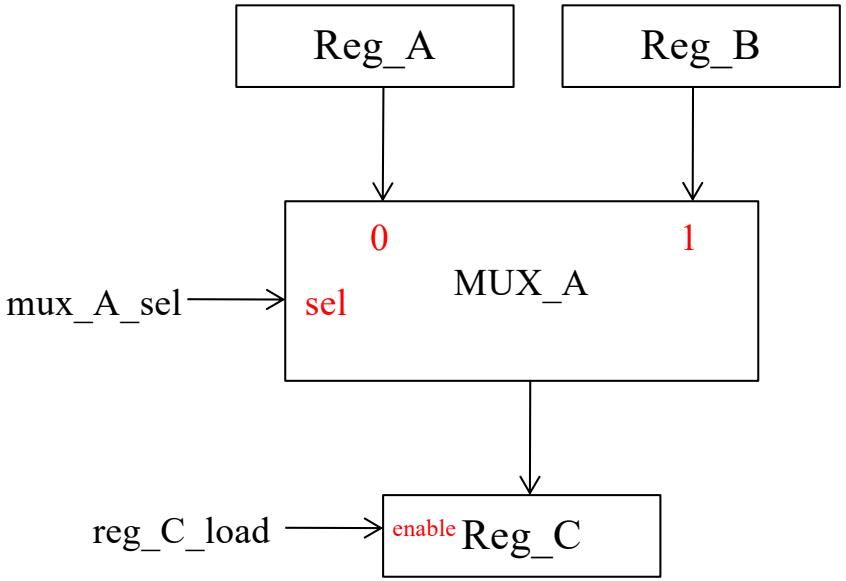
Micro-ROM format



Block Diagram Design

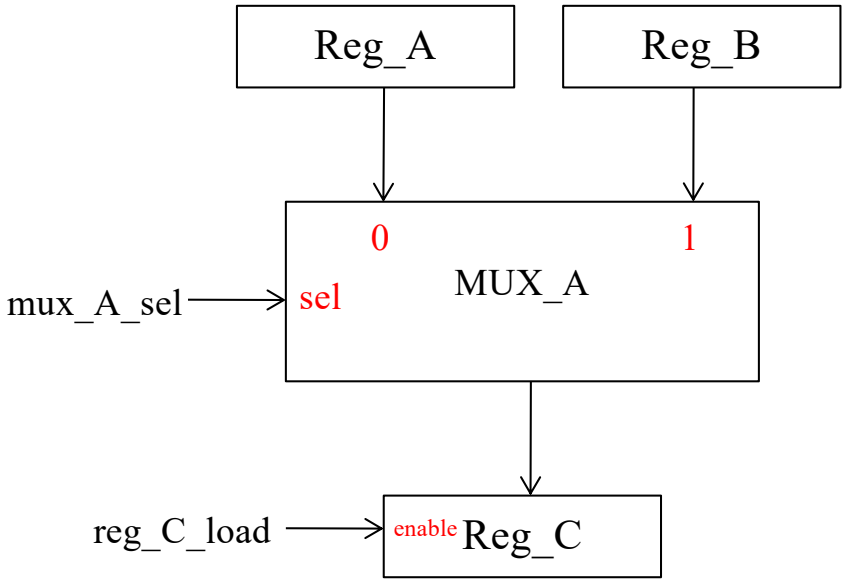
- Look at the set of RTL statements and group them according to destination

1. Reg_C ⟵ Reg_A
2. Reg_C ⟵ Reg_B



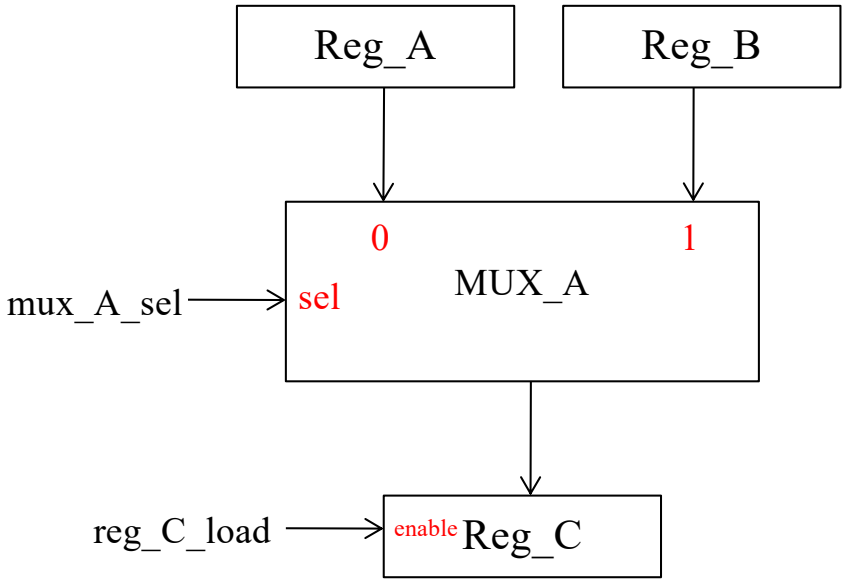
Control Signals

- mux_A_sel
- reg_C_load



Microcode

- Reg_C ⟵ Reg_A
  - mux_A_sel = 0
  - reg_C_load = 1
- Reg_C ⟵ Reg_B
  - mux_A_sel = 1
  - reg_C_load = 1



RTL	Micro-operations	
	mux_A_sel	reg_C_load
Reg_C ⟵ Reg_A	0	1
Reg_C ⟵ Reg_B	1	1

uROM Content - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**This is a sample microcode worksheet**															
2	**The control signals shown are examples and your signals should be difference**															
3		uROM address	PC clear	PC inc	PC load	MDR load	MDR mux select	R0 mux select	MAR load	MAR mux select	MEM write	IR load	Z load	cond	Map	Next_Addr
4	NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001
5	IF1: MAR <- PC	1	0	0	0	0	0	0	1	1	0	0	0	0	0	00000010
6	IF2: MDR <- M[MAR], PC<-PC+1	2	0	1	0	1	0	0	0	0	0	0	0	0	0	00000011
7	IF3: IR <- MDR	3	0	0	0	0	0	0	0	0	0	1	0	0	1	00000000
8	LOADI1: MAR<- PC	10	0	0	0	0	0	0	1	1	0	0	0	0	0	00010001
9																
10																
11	LOAD1: MAR<- PC	20	0	0	0	0	0	0	1	1	0	0	0	0	0	00100001
12																

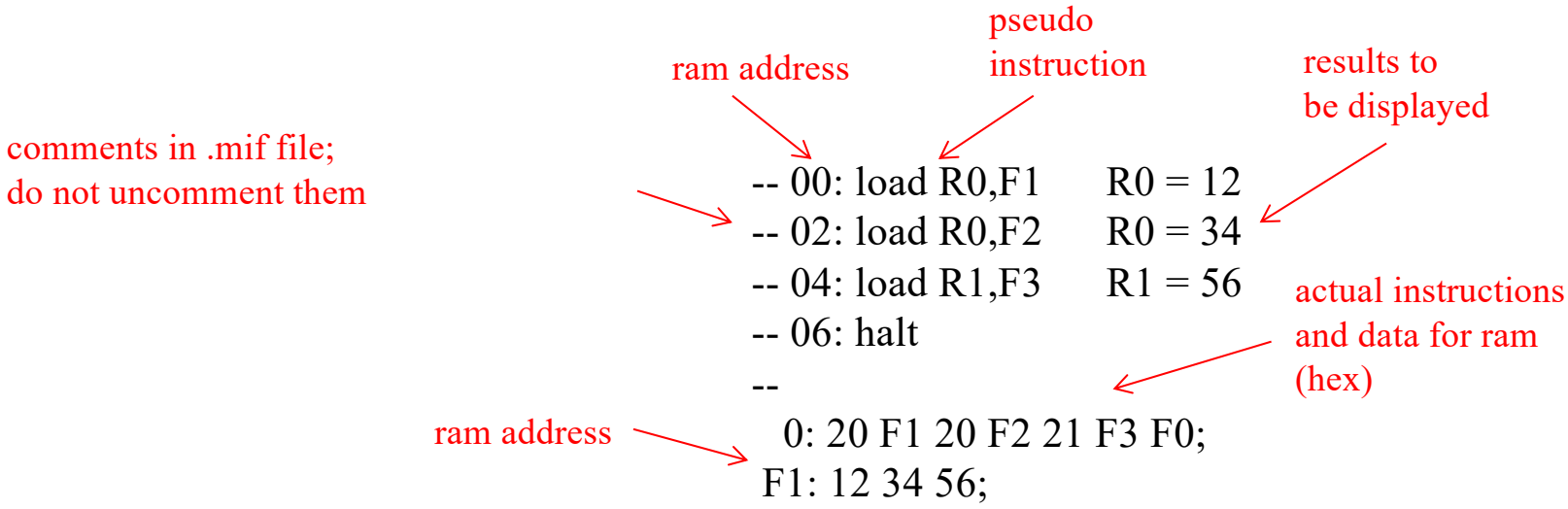
```
-- ECE 495 Exp. 7
-- uROM content sample
-- Dr. Hou
--
depth = 256;
width = 21;
address_radix = hex;
data_radix = bin;
content

--           PC clear
--           |PC inc
--           ||PC load
--           |||MDR load
--           ||||MDR mux select
--           |||||R0 mux select
--           ||||||MAR Load
--           |||||||MAR mux select
--           ||||||||MEM Write
--           |||||||||IR load
--           ||||||||||Z load
--           |||||||||||cond
--           ||||||||||||
  begin   --123456789012876543210
[0..FF]:    000000000000000000;
                                      -- IF
    0:      000000000000000001;  --
    1:      000001100000000010;  -- MAR<-PC
    2:      010010000000000011;  -- MDR<-M[MAR], PC<-PC+1
    3:      000001100001000001   -- IR<-MDR, map

   10:      000001100000000010001;

   20:      000001100000000100001;

-- more lines
  end;
```

Displaying Results

- Display the content of R0, R1
- Display the content of PC
- Display the content of SP
- Display the content of Z (use LED)
- Addition display of other registers maybe needed for debugging.
- You can use seven segment or LCD for display

Program RAM File

- A number of test programs will be provided to test your CPU



Mnemonic	Code (Binary)	Code (HEX)
nop	00000000	00
loadi R0,FF	00010000 11111111	10 FF
loadi R1,FF	00010001 11111111	11 FF
load R0,FF	00100000 11111111	20 FF
load R1,FF	00100001 11111111	21 FF
store FF,R0	00110000 11111111	30 FF
store FF,R1	00110001 11111111	31 FF
move r0,r1	01000000	40
move r1,r0	01000001	41
add R0,R1	01010000	50
add R1,R0	01010001	51
sub R0,R1	01100000	60
sub R1,R0	01100001	61

Mnemonic	Code (Binary)	Code (Hex)
testnz R0	01110000	70
testnz R1	01110001	71
testz R0	10000000	80
testz R1	10000001	81
jump FF	10010000 11111111	90 FF
jumpz FF	10100000 11111111	A0 FF
loadsp FF	10110000 11111111	B0 FF
peep R0	11000000	C0
peep R1	11000001	C1
push R0	11010000	D0
push R1	11010000	D1
pop R0	11100000	E0
pop R1	11100001	E1
halt	11110000	F0

Test 1 - LOADI

```
-- 00: loadi R0,47       R0 = 47
-- 02: loadi R1,F2       R0 = 47, R1 = F2
-- 04: loadi R0,8F       R0 = 8F, R1 = F2
-- 06: loadi R1,FF       R0 = 8F, R1 = FF
-- 08: halt
--
00: 10 47 11 F2 10 8F 11 FF F0;
```

Mnemonic	Code (Binary)	Code (HEX)
nop	00000000	00
loadi R0,FF	00010000 11111111	10 FF
loadi R1,FF	00010001 11111111	11 FF
load R0,FF	00100000 11111111	20 FF
load R1,FF	00100001 11111111	21 FF
store FF,R0	00110000 11111111	30 FF
store FF,R1	00110001 11111111	31 FF
move r0,r1	01000000	40
move r1,r0	01000001	41
add R0,R1	01010000	50
add R1,R0	01010001	51
sub R0,R1	01100000	60
sub R1,R0	01100001	61

Test 2 - LOAD

```
-- 00: load R0, B3        R0 = 8D
-- 02: load R1, B5        R0 = 8D, R1 = FE
-- 04: load R0, D1        R0 = AE, R1 = FE
-- 06: load R1, D2        R0 = AE, R1 = 2F
-- 08: halt
--
00: 20 B3 21 B5 20 D1 21 D2 F0;
B3: 8D 00 FE;
D1: AE 2F;
```

Mnemonic	Code (Binary)	Code (HEX)
nop	00000000	00
loadi R0,FF	00010000 11111111	10 FF
loadi R1,FF	00010001 11111111	11 FF
load R0,FF	00100000 11111111	20 FF
load R1,FF	00100001 11111111	21 FF
store FF,R0	00110000 11111111	30 FF
store FF,R1	00110001 11111111	31 FF
move r0,r1	01000000	40
move r1,r0	01000001	41
add R0,R1	01010000	50
add R1,R0	01010001	51
sub R0,R1	01100000	60
sub R1,R0	01100001	61