Computer Architecture Lab

Lab08 – Week #8

MPLAB
Media Processing Laboratory

컴퓨터 공학과 이성원 교수님 연구실

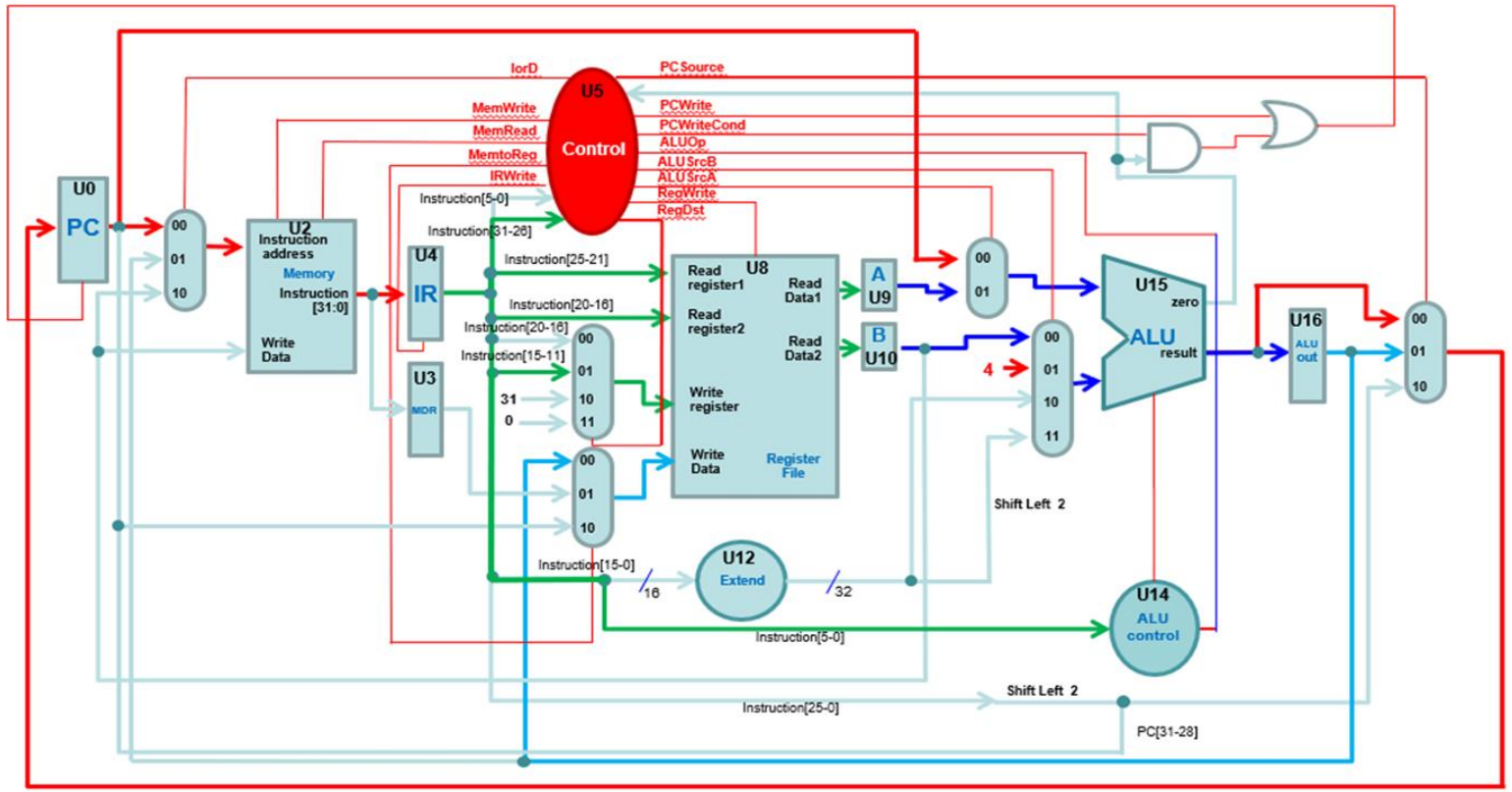
R-type (SUBU)

0_1_0_000_1_xx_xxx_0_x_011_001_00100_00_000_00_1_xxxxxxxx_11 // 0x00: FETCH

x_x_0_xxx_0_xx_xxx_0_1_011_100_00100_00_xxx_xx_0_xxxxxxxx_01 // 0x01: DECODE/REG_READ/BRANCH_ADDR

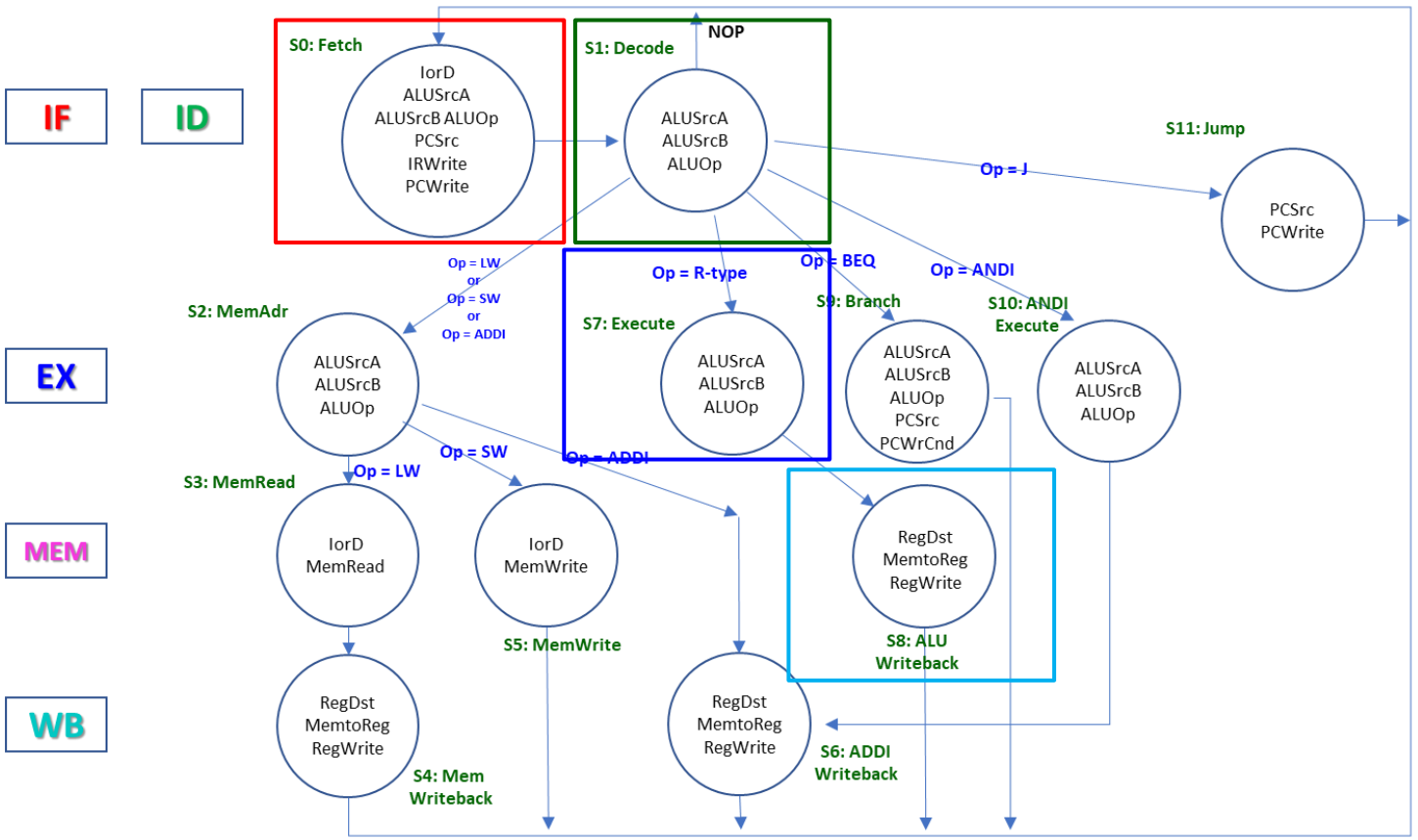
x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_xxxxxxxx_11 // 0x02: subu Excution

x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_xxxxxxxx_00 // 0x03: subu R-type completion



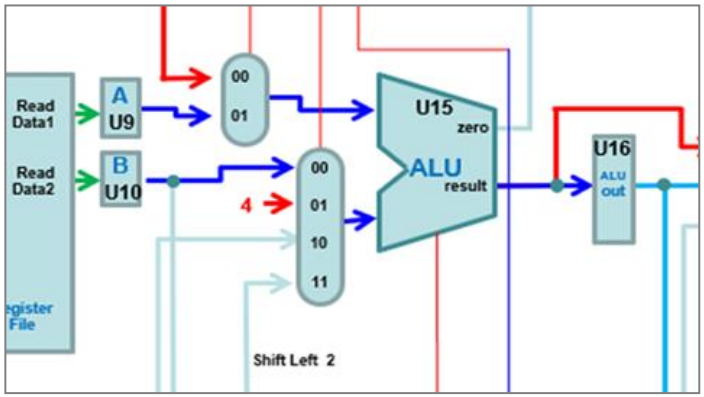
R-type (SUBU)

```
0_1_0_000_1_xx_xxx_0_x_011_001_00100_00_000_00_1_xxxxxxxx_11  // 0x00:  FETCH
x_x_0_xxx_0_xx_xxx_0_1_011_100_00100_00_xxx_xx_0_xxxxxxxx_01  // 0x01:  DECODE/REG_READ/BRANCH_ADDR
x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_xxxxxxxx_11  // 0x02:  subu Excution
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_xxxxxxxx_00  // 0x03:  subu R-type completion
```

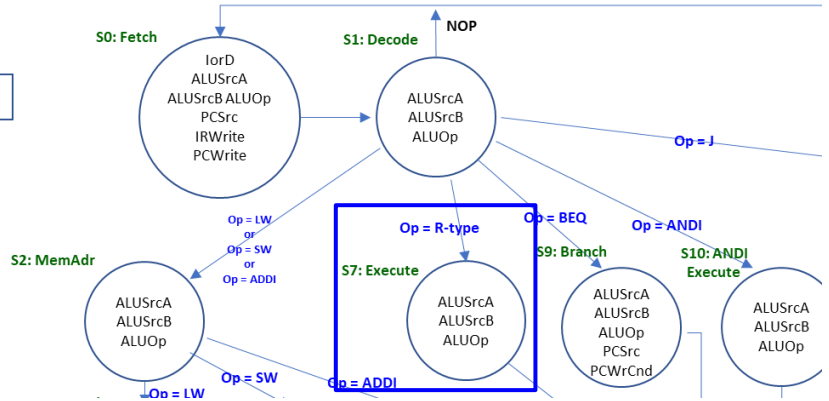


SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID EX



x

IorD	Memory Access for Instruction or Data
0	Instruction
1	Data

x

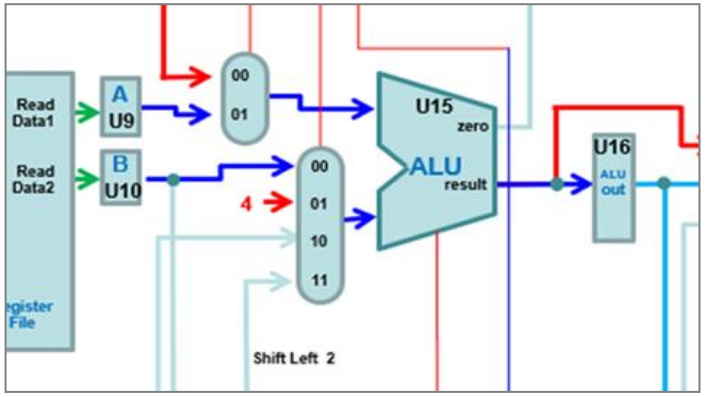
MemRead	Memory Read Access Enable
0	No Memory Read Access
1	Memory Read Access

0

MemWrite	Memory Write Access Enable
0	No Memory Write
1	Memory Write Enable

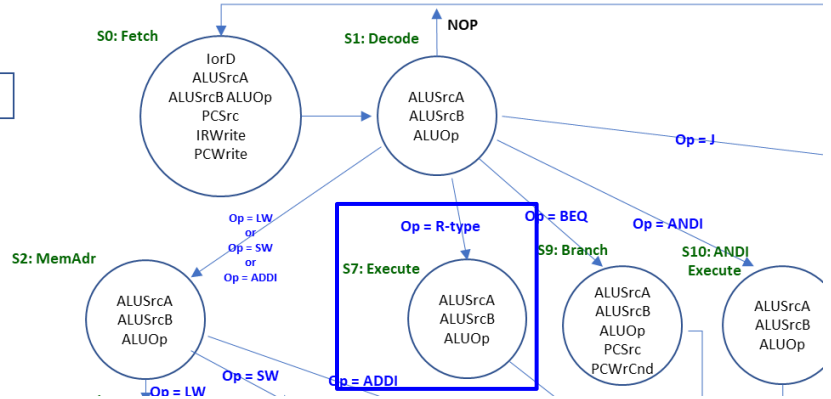
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX



XXX

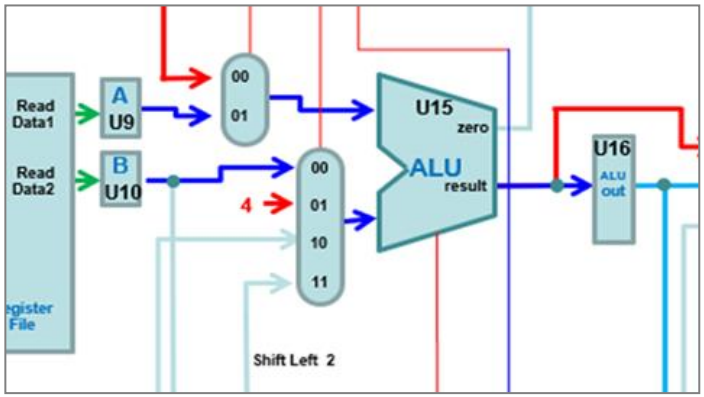
DatWidth	Data Width for Memory Access
000	32-bit Word
010	16-bit Halfword
011	8-bit Byte
110	16-bit Halfword /w Sign Ext
111	8-bit Byte /w Sign Ext

0

IRwrite	Instruction Register Write Enable
0	No Instruction Register Write
1	Instruction Register Write Enable

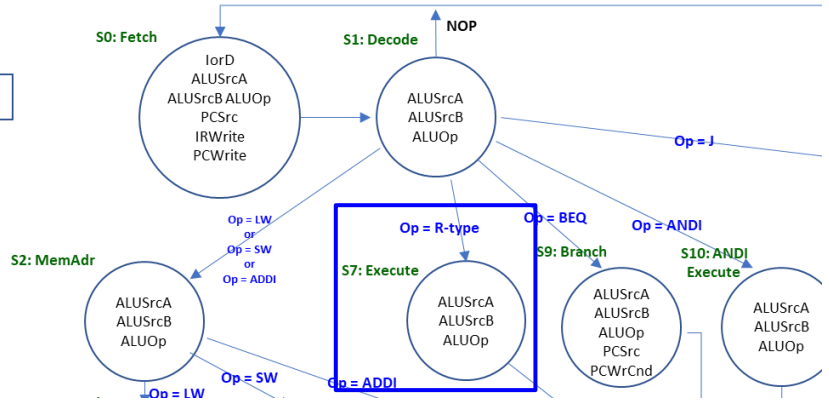
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX



XX

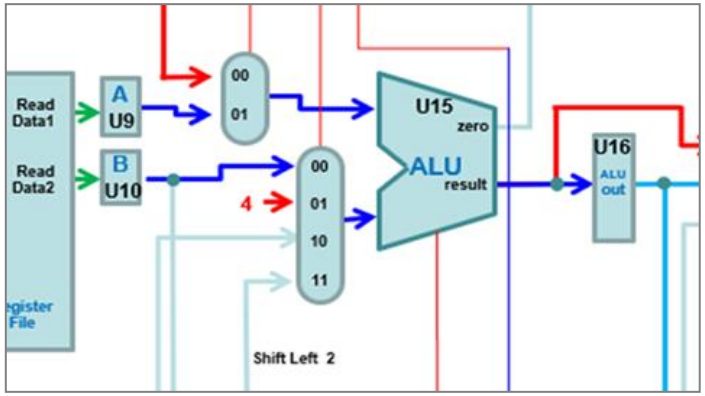
RegDst	Selection for Destination Register
00	Write to $rt
01	Write to $rd
10	Write to $rs
11	Write to $31

XXX

RegDatSel	To Select Data Source for Register Write
000	Write ALUOut to Register file
001	Write MDR to Register file
010	Write LO to Register file
011	Write HI to Register file
100	Write PC to Register file

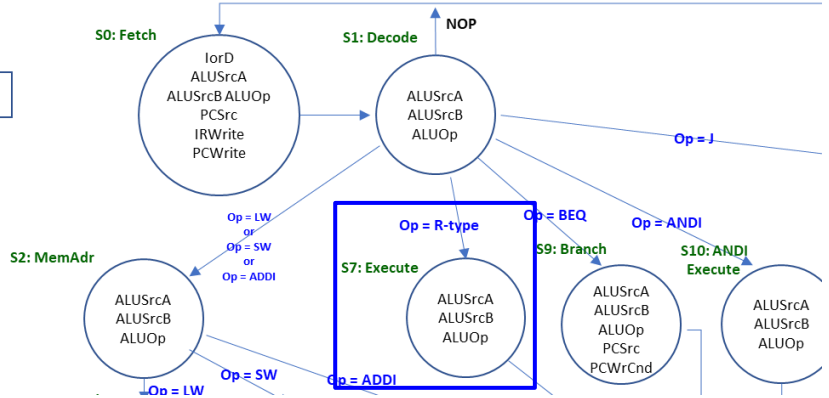
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX



0

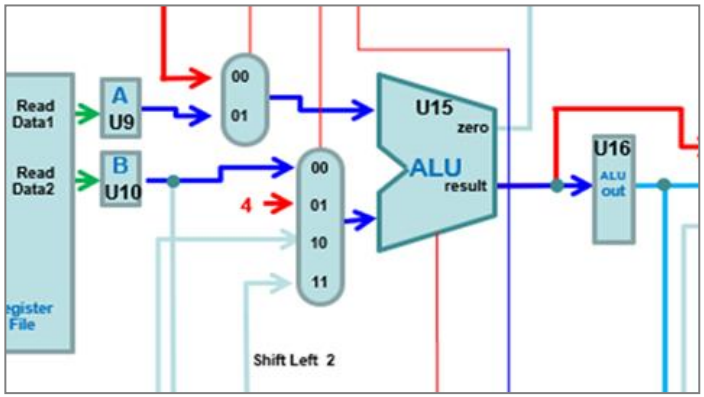
RegWrite	Write Enable to Register File
0	Do not write to Register file
1	Write to Register file

X

EXTmode	Immediate Data Extension Mode
0	Zero Extension
1	Sign Extension

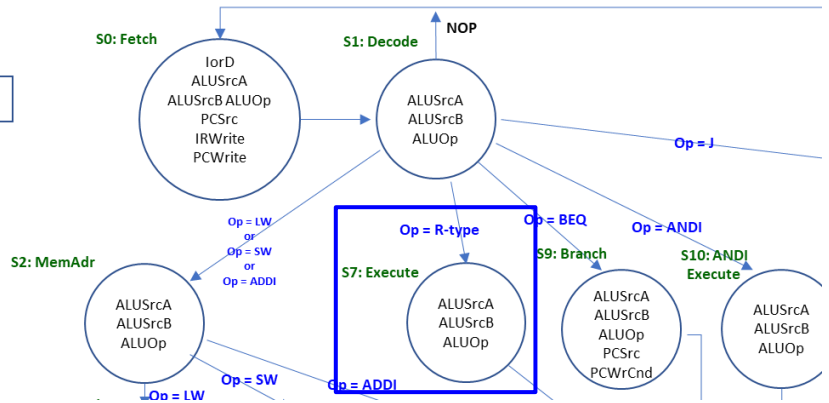
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX



000

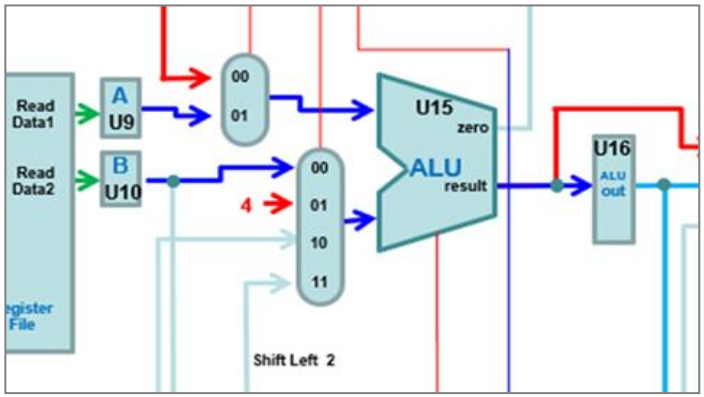
ALUsrcA	ALU Input A Source Selection
000	Register A to ALU input A
001	0x4
010	0x0
011	PC to ALU input A
100	MDR to ALU input A

000

ALUsrcB	ALU Input B Source Selection
000	Register B to ALU input B
001	0x4
010	0x0
011	SEU output to ALU input B
100	SEU output << 2

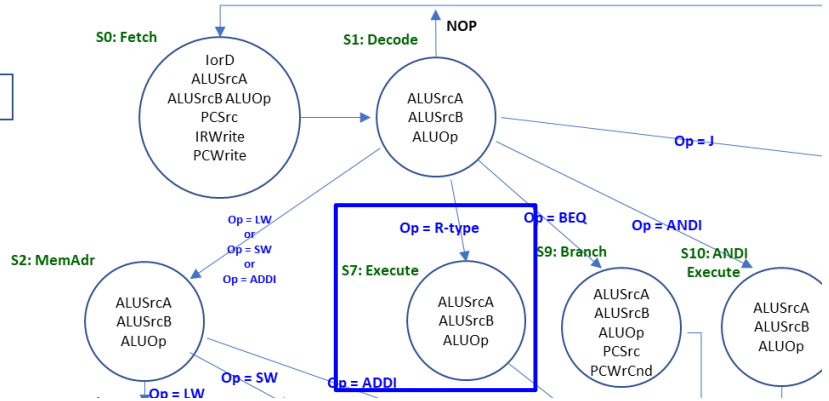
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

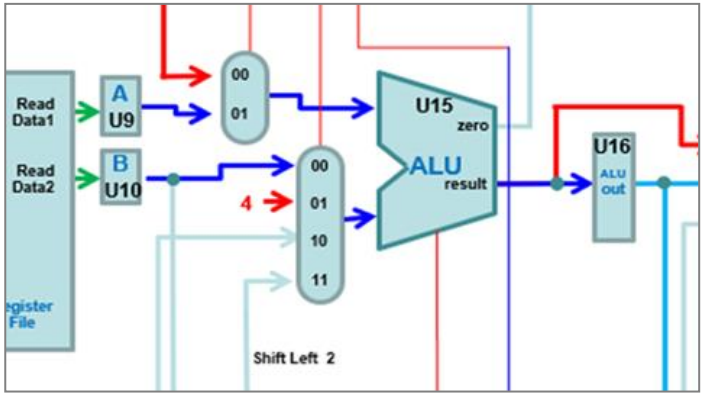
EX



ALUop	ALU Operation Code		
00000	Bitwise AND	01001	a × b
00001	Bitwise OR	01010	Unsigned a × b
00010	Bitwise NOR	01011	a / b
00011	Bitwise XOR	01100	Unsigned a / b
00100	a + b	01101	b << a
00101	Unsigned a + b	01110	b >> a
00110	a – b	01111	b >>> a
00111	Unsigned a – b	10000	Set Less Than
01000	Zero	10001	Unsigned SLT
10010	HI = a	10011	LO = a

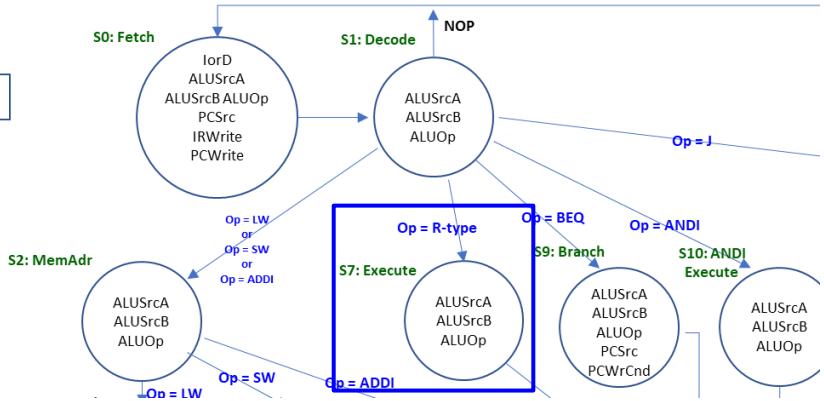
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX

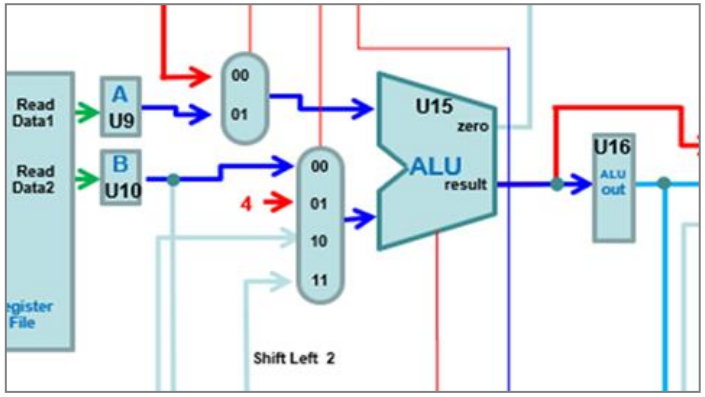


0x

ALUctrl[1:0]	Extra ALU Control Signal
ALUctrl[0]=0	Shift = Shift Amount
ALUctrl[0]=1	Shift = $rs
ALUctrl[1]=0	Normal ALU input (a,b)
ALUctrl[1]=1	Exchanged ALU input (b,a)

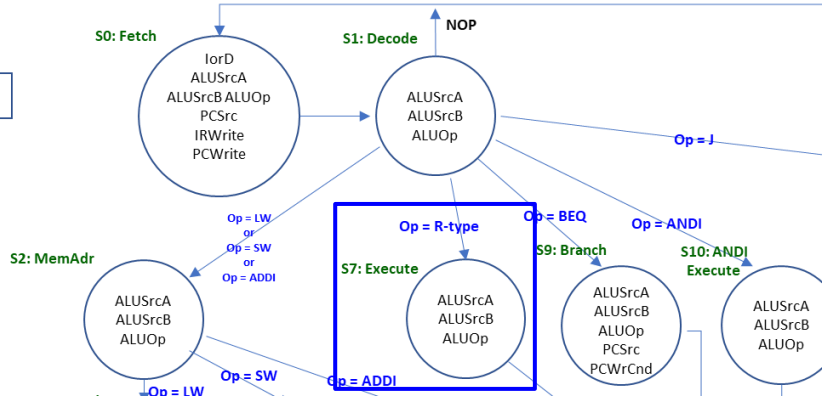
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX

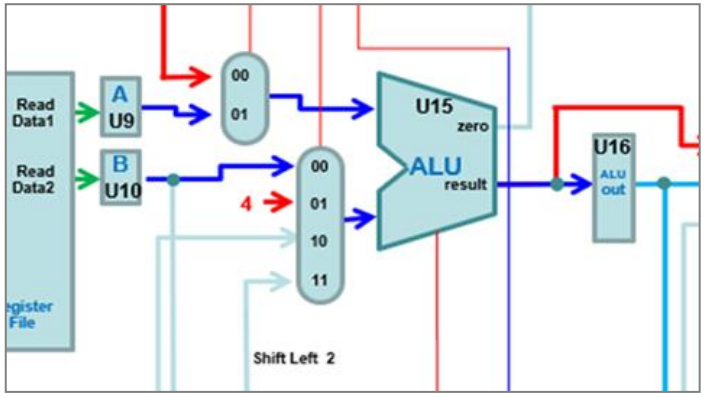


XXX

Branch	Branch Options
000	No branch condition or Jump
001	Reserved
010	Branch if not negative
011	Branch if negative
100	Branch if equal
101	Branch if not equal
110	Branch if not positive
111	Branch if positive

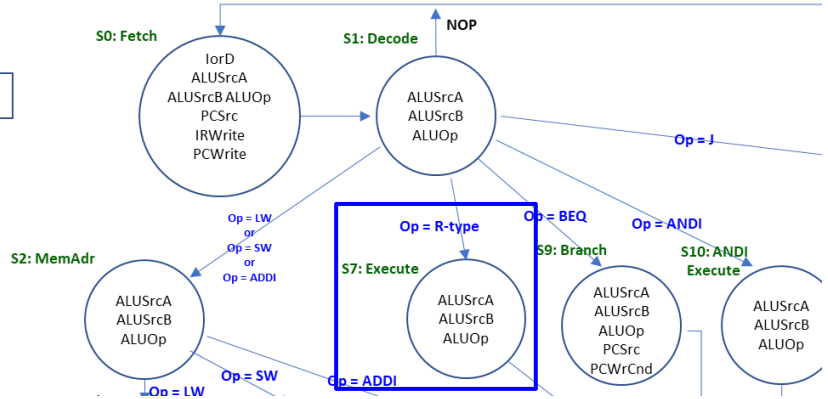
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX



XX

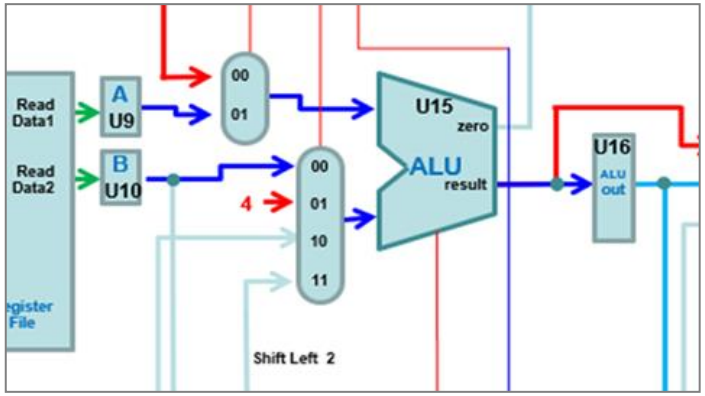
PCsrc	PC Data Source Selection
00	From ALU output
01	From ALUOut Register
10	From Jump Address
11	From Current PC

0

PCwrite	PC Register Write Enable
0	No PC Register Write
1	PC Register Write Enable

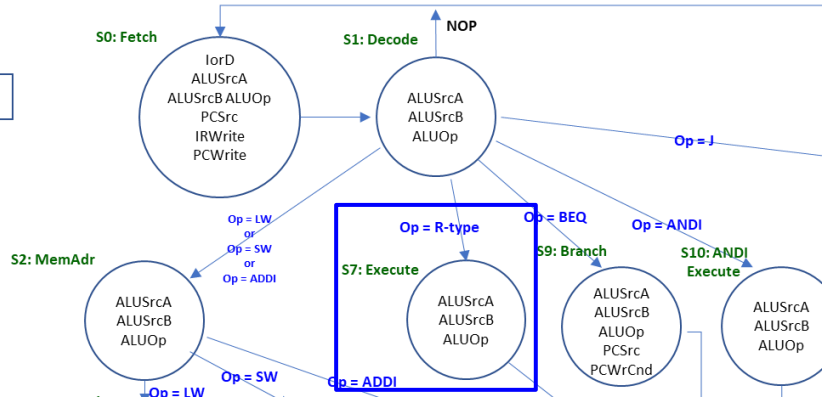
SUBU [execution] ($d = $s - $t)

x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11 //0x05 : subu



IF ID

EX



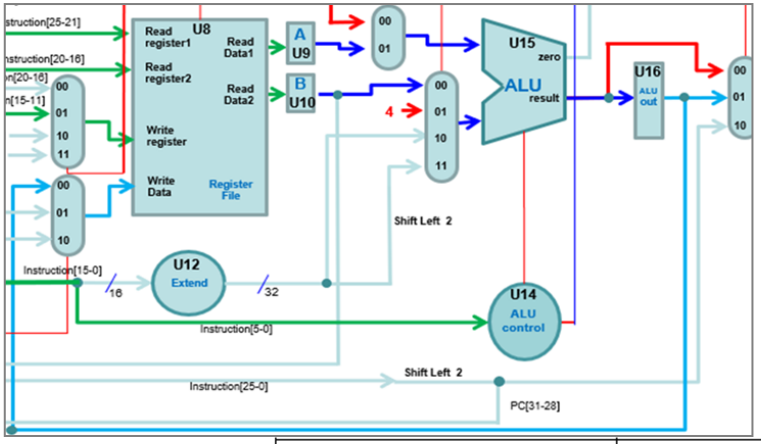
11

StateSel	Next State Selection Signal
00	Next State = 0
01	Next State = State Indicated by Instruction
10	Reserved
11	Next State = Current State + 1

*** 8 bits before StateSel signal are reserved. They should be set xxxxxxxx.**

SUBU [write back] ($d = $s - $t)

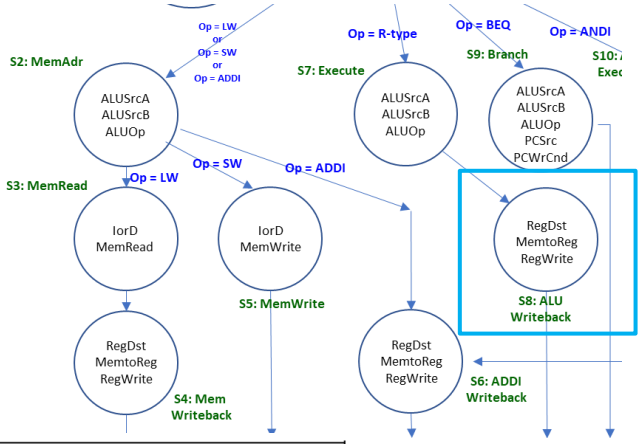
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB



x

IorD	Memory Access for Instruction or Data
0	Instruction
1	Data

x

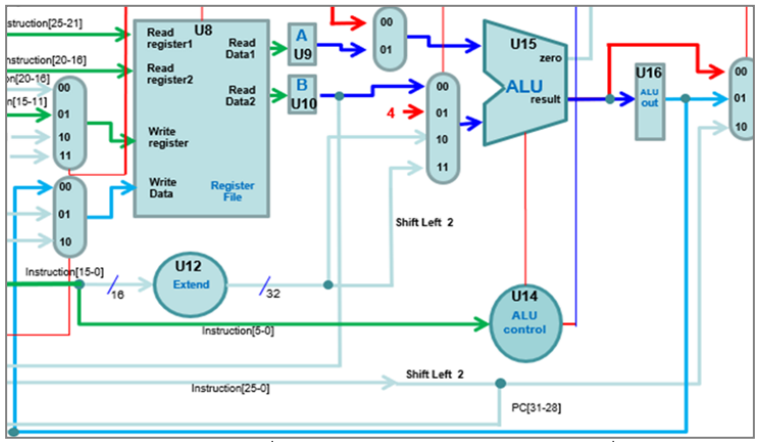
MemRead	Memory Read Access Enable
0	No Memory Read Access
1	Memory Read Access

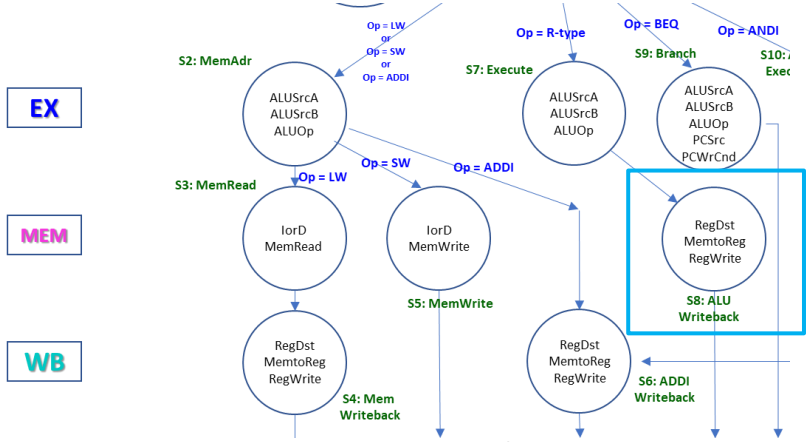
0

MemWrite	Memory Write Access Enable
0	No Memory Write
1	Memory Write Enable

SUBU [write back] ($d = $s - $t)

x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu





xxx

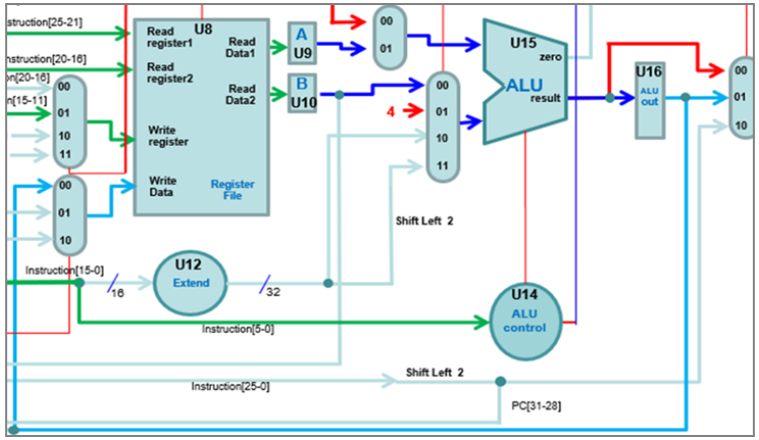
DatWidth	Data Width for Memory Access
000	32-bit Word
010	16-bit Halfword
011	8-bit Byte
110	16-bit Halfword /w Sign Ext
111	8-bit Byte /w Sign Ext

0

IRwrite	Instruction Register Write Enable
0	No Instruction Register Write
1	Instruction Register Write Enable

SUBU [write back] ($d = $s - $t)

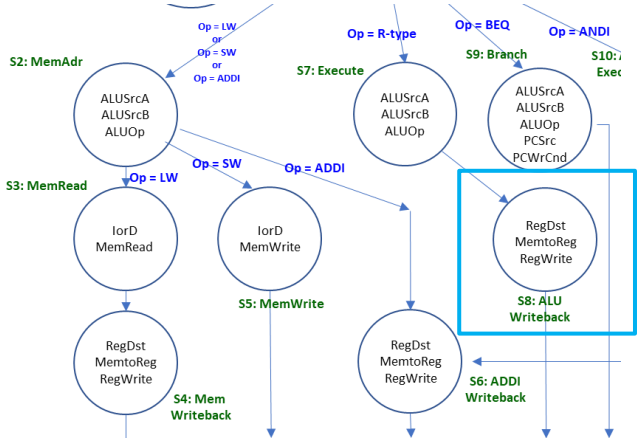
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB



01

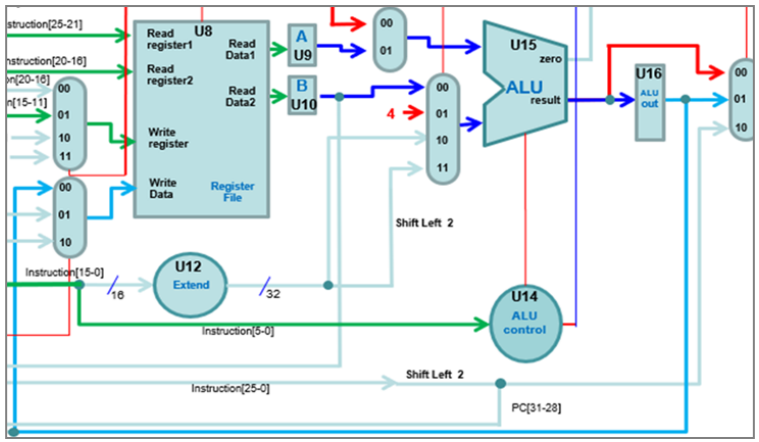
RegDst	Selection for Destination Register
00	Write to $rt
01	Write to $rd
10	Write to $rs
11	Write to $31

000

RegDatSel	To Select Data Source for Register Write
000	Write ALUOut to Register file
001	Write MDR to Register file
010	Write LO to Register file
011	Write HI to Register file
100	Write PC to Register file

SUBU [write back] ($d = $s - $t)

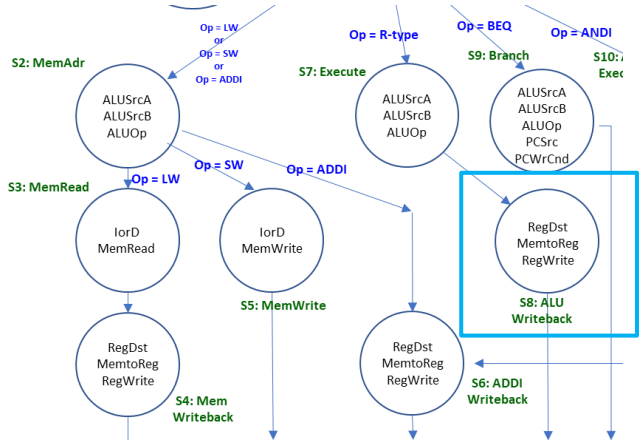
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB



1

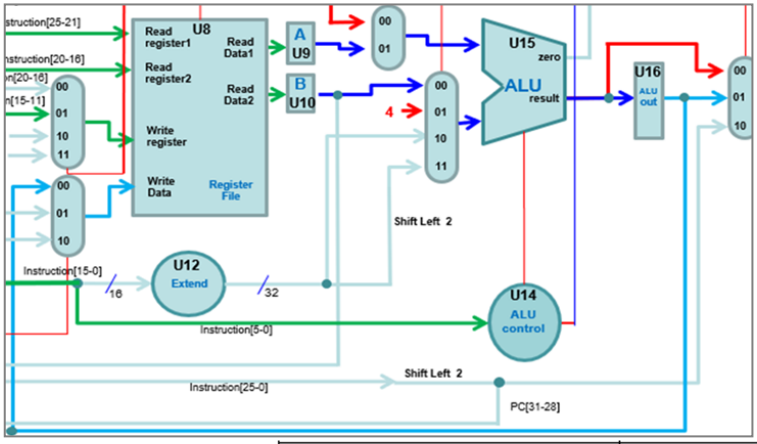
RegWrite	Write Enable to Register File
0	Do not write to Register file
1	Write to Register file

X

EXTmode	Immediate Data Extension Mode
0	Zero Extension
1	Sign Extension

SUBU [write back] ($d = $s - $t)

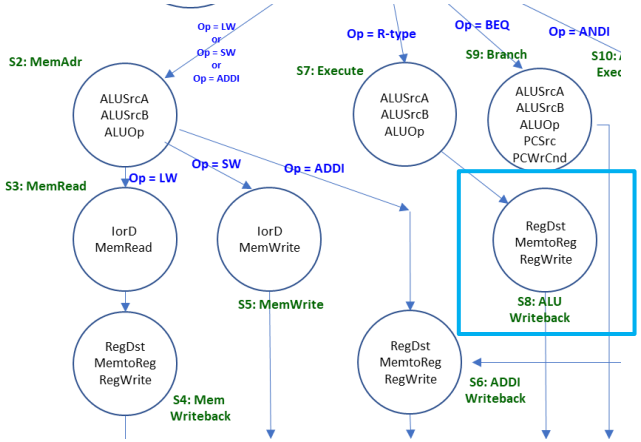
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB



XXX

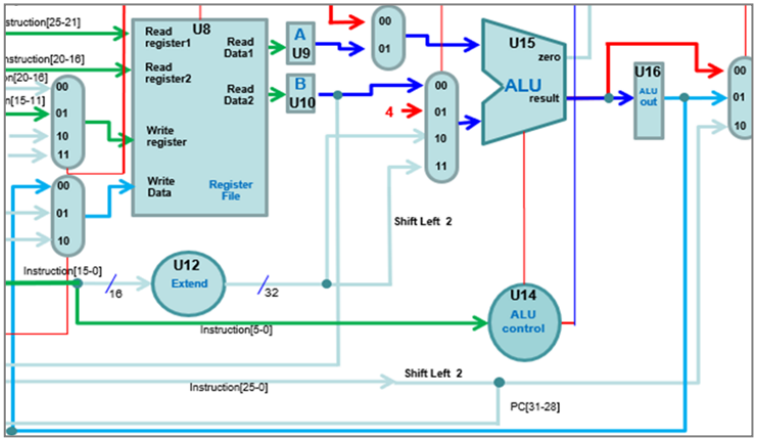
ALUsrcA	ALU Input A Source Selection
000	Register A to ALU input A
001	0x4
010	0x0
011	PC to ALU input A
100	MDR to ALU input A

XXX

ALUsrcB	ALU Input B Source Selection
000	Register B to ALU input B
001	0x4
010	0x0
011	SEU output to ALU input B
100	SEU output << 2

SUBU [write back] ($d = $s - $t)

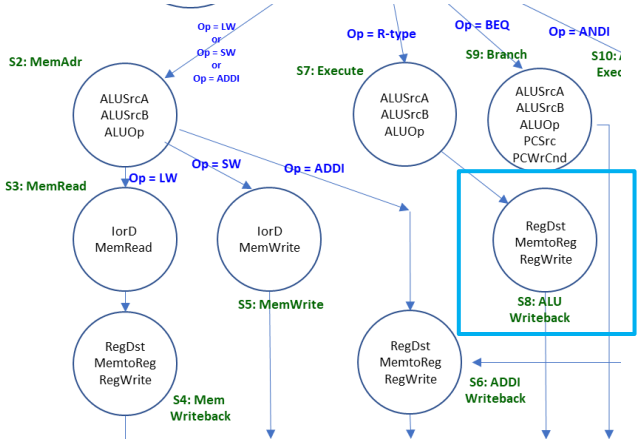
x_x_0_xxx_0_01_000_1_x_xxx_xxx_XXXXX_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB

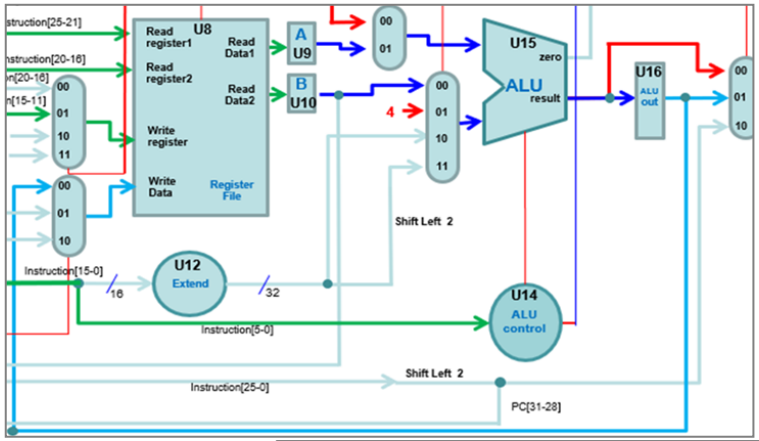


XXXXX

ALUop	ALU Operation Code		
00000	Bitwise AND	01001	a × b
00001	Bitwise OR	01010	Unsigned a × b
00010	Bitwise NOR	01011	a / b
00011	Bitwise XOR	01100	Unsigned a / b
00100	a + b	01101	b << a
00101	Unsigned a + b	01110	b >> a
00110	a – b	01111	b >>> a
00111	Unsigned a – b	10000	Set Less Than
01000	Zero	10001	Unsigned SLT
10010	HI = a	10011	LO = a

SUBU [write back] ($d = $s - $t)

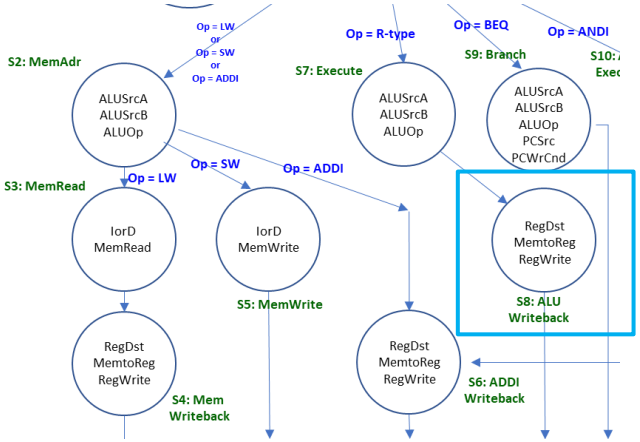
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB

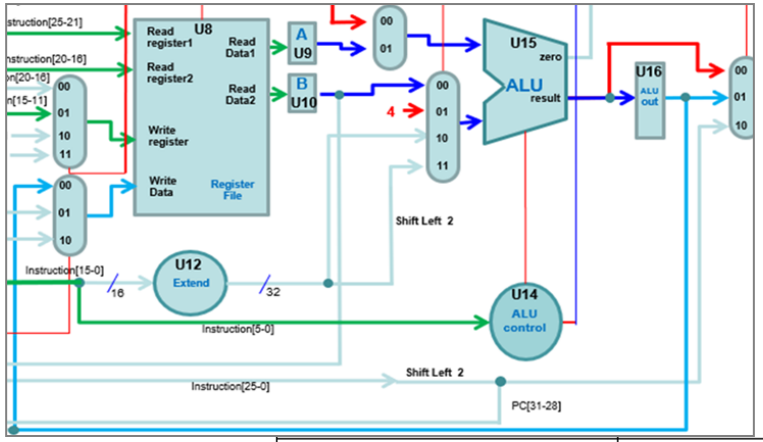


XX

ALUctrl[1:0]	Extra ALU Control Signal
ALUctrl[0]=0	Shift = Shift Amount
ALUctrl[0]=1	Shift = $rs
ALUctrl[1]=0	Normal ALU input (a,b)
ALUctrl[1]=1	Exchanged ALU input (b,a)

SUBU [write back] ($d = $s - $t)

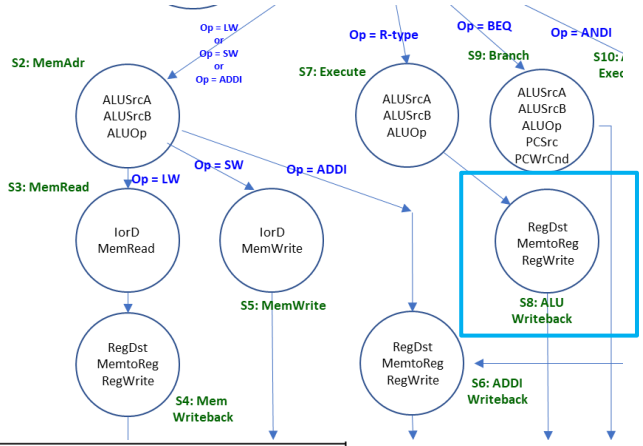
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB

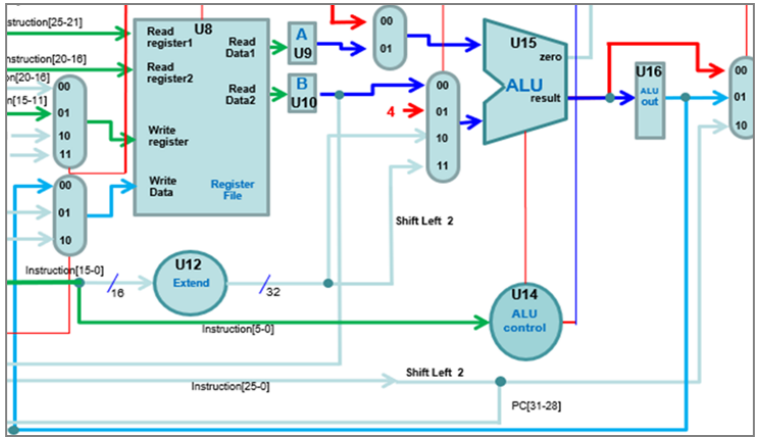


XXX

Branch	Branch Options
000	No branch condition or Jump
001	Reserved
010	Branch if not negative
011	Branch if negative
100	Branch if equal
101	Branch if not equal
110	Branch if not positive
111	Branch if positive

SUBU [write back] ($d = $s - $t)

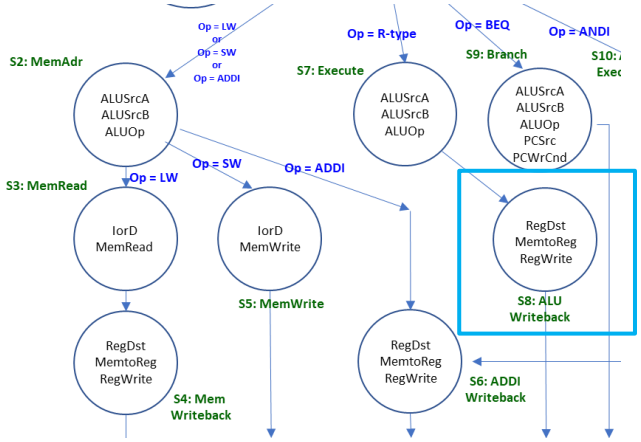
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB



XX

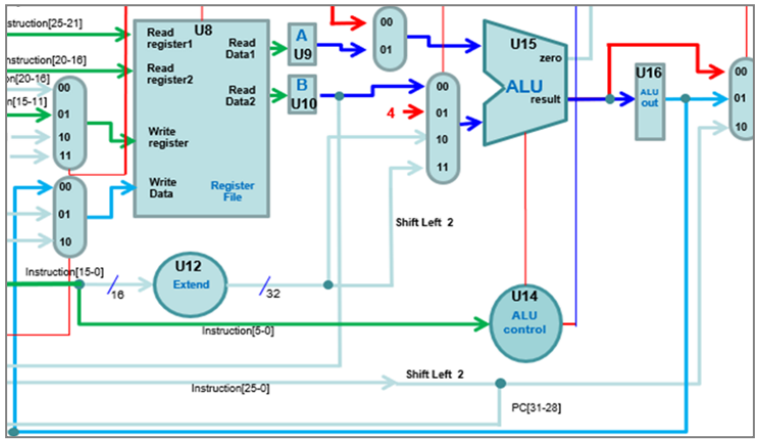
PCsrc	PC Data Source Selection
00	From ALU output
01	From ALUOut Register
10	From Jump Address
11	From Current PC

0

PCwrite	PC Register Write Enable
0	No PC Register Write
1	PC Register Write Enable

SUBU [write back] ($d = $s - $t)

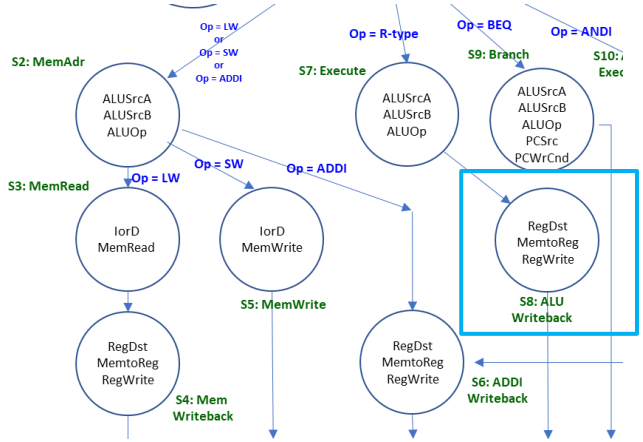
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00 //0x06 : subu



EX

MEM

WB



00

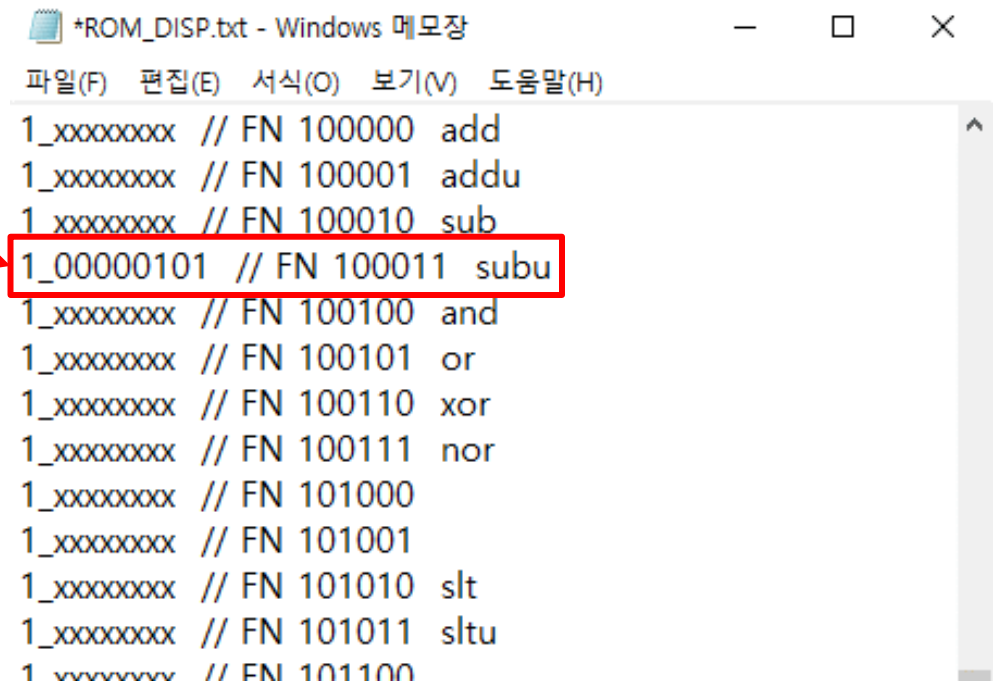
StateSel	Next State Selection Signal
00	Next State = 0
01	Next State = State Indicated by Instruction
10	Reserved
11	Next State = Current State + 1

*** 8 bits before StateSel signal are reserved. They should be set xxxxxxxx.**

SUBU – ROM_DISP.txt

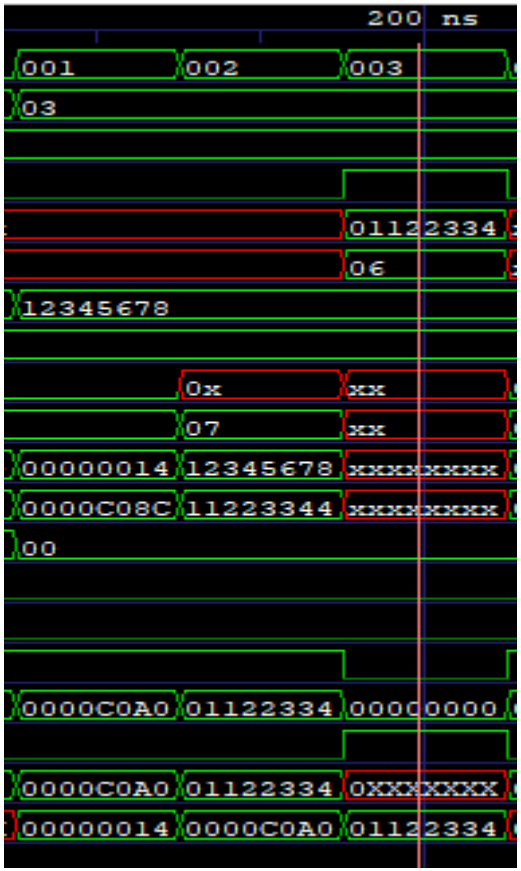
```
x_x_0_xxx_0_xx_xxx_0_x_000_000_00111_0x_xxx_xx_0_11      //0x05 : subu
x_x_0_xxx_0_01_000_1_x_xxx_xxx_xxxxx_xx_xxx_xx_0_00      //0x06 : subu
```

Address for ROM_MICRO



SUBU – Waveform

```
      o_state[8:0] =003
  i_Read_regl[4:0] =03
  i_Read_reg2[4:0] =05
        i_RegWrite =1
i_Write_data[31:0] =01122334
  i_Write_reg[4:0] =06
o_Read_datal[31:0] =12345678
o_Read_data2[31:0] =11223344
    i_ALUctrl[1:0] =xx
      i_ALUop[4:0] =xx
     i_datal[31:0] =xxxxxxxx
     i_data2[31:0] =xxxxxxxx
      i_shamt[4:0] =00
           o_carry =0
        o_overflow =0
        o_positive =0
    o_result[31:0] =00000000
            o_zero =1
          i_x[31:0] =0XXXXXXX
          o_y[31:0] =01122334
```



Thank You