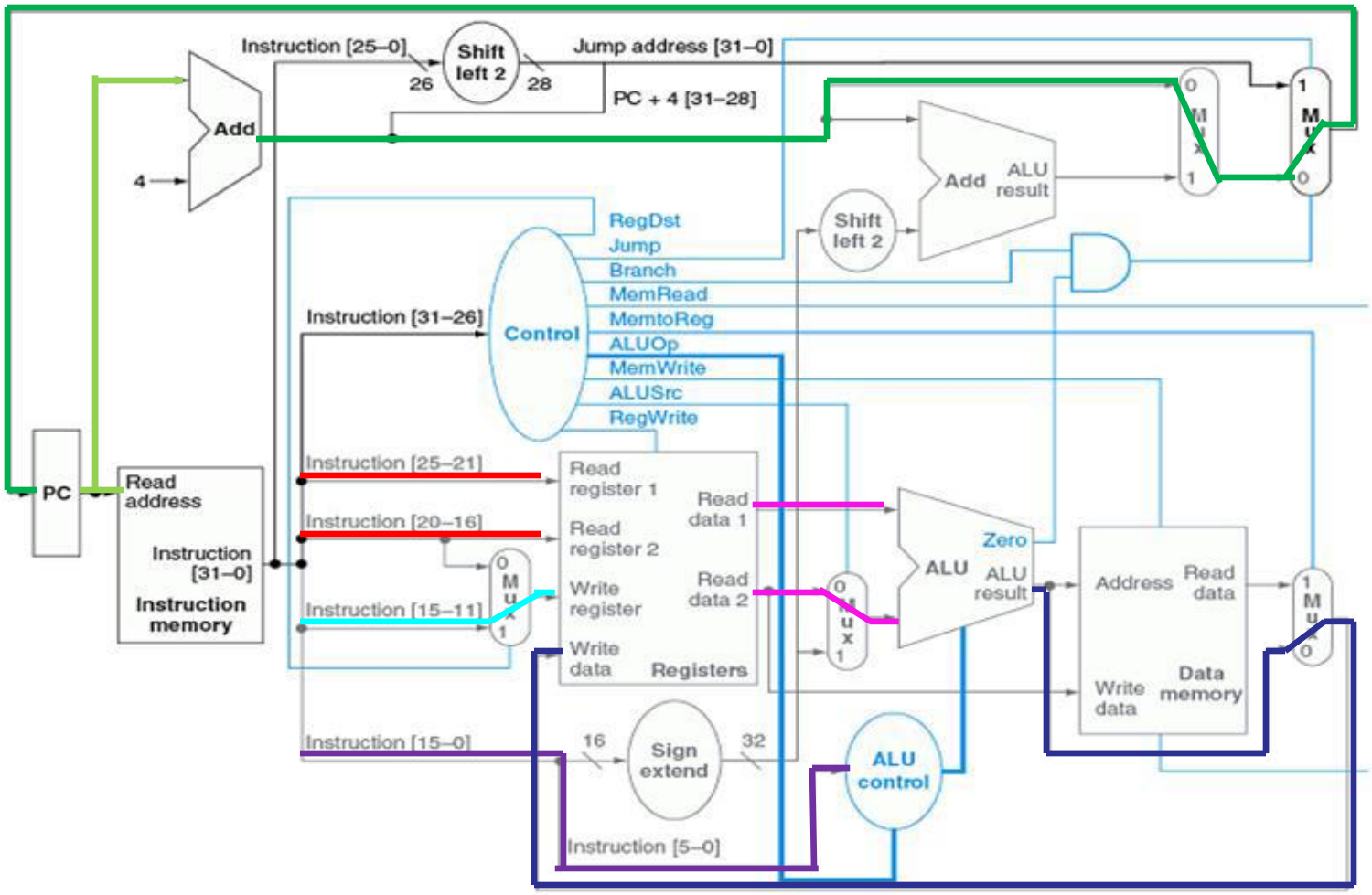
Computer Architecture Lab

Lab05 – Week #5

MPLAB Media Processing Laboratory

컴퓨터 공학과 이성원 교수님 연구실

Single Cycle CPU_ex.subu



Single Cycle CPU module

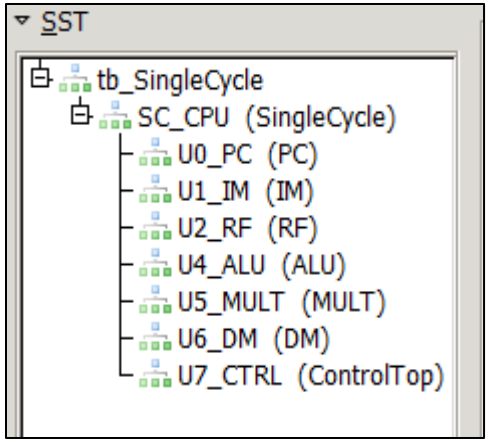


Table 1 – Instance name of top module

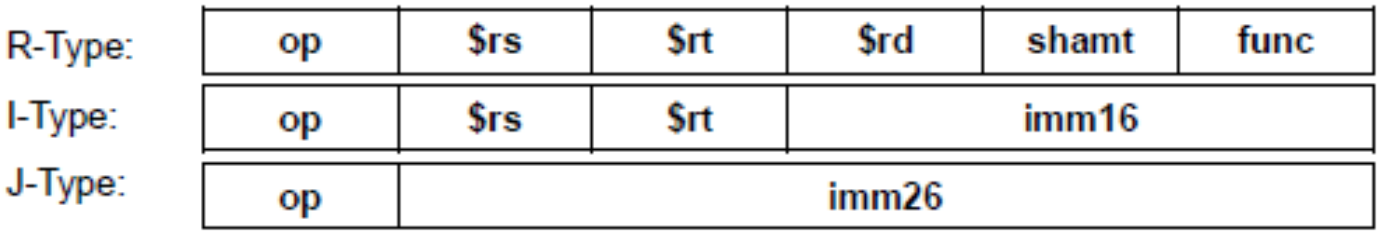
Instance name	Description
SC_CPU	Top module. Single-Cycle CPU
U0_PC	Program counter
U1_IM	Instruction memory
U2_RF	Register file
U3_SEU	Sign Extension Unit
U4_ALU	Arithmetic Logic Unit
U5_MULT	Multiplier Logic Unit
U6_DM	Data memory
U7_CTRL	Control unit -MainControl, MyControl

PLA_AND.txt

Table 2 – Consecutive Instruction Decoding Configuration in PLA_AND.txt

Port name	Classification	Bit	Description
Op	Input	6-bit	Op code
Func	Input	6-bit	Function code
Regimm	Input	5-bit	Register Immediate code (RT)

MIPS Inst Datasheet.pdf



PLA_AND.txt

- R-type's opcode is 000000
- When implementing R-type instructions, you should fill Function
- RegImm is filled when you implement specific I-type instructions (ex.bltz, bgez)

RegImm (I-Type) with $rt:

H / L	000	001	010	011	100	101	110	111
00	bltz	bgez	bltzl	bgezl				
01	tgei	tgeiu	tlti	tltiu	teqi		tnei	
10	bltzal	bgezal	bltzall	bgezall				
11								

subu	100011	f $d, $s, $t	$d = $s - $t

```
xxxxxx_xxxxxx_xxxxx  // 0x00 : sll
xxxxxx_xxxxxx_xxxxx  // 0x01 : srl
xxxxxx_xxxxxx_xxxxx  // 0x02 : sra
xxxxxx_xxxxxx_xxxxx  // 0x03 : sllv
xxxxxx_xxxxxx_xxxxx  // 0x04 : srlv
xxxxxx_xxxxxx_xxxxx  // 0x05 : srav
xxxxxx_xxxxxx_xxxxx  // 0x06 : jr
xxxxxx_xxxxxx_xxxxx  // 0x07 : jalr
xxxxxx_xxxxxx_xxxxx  // 0x08 : break
xxxxxx_xxxxxx_xxxxx  // 0x09 : mfhi
xxxxxx_xxxxxx_xxxxx  // 0x0a : mthi
xxxxxx_xxxxxx_xxxxx  // 0x0b : mflo
xxxxxx_xxxxxx_xxxxx  // 0x0c   mtlo
xxxxxx_xxxxxx_xxxxx  // 0x0d   mult
xxxxxx_xxxxxx_xxxxx  // 0x0e   multu
xxxxxx_xxxxxx_xxxxx  // 0x0f   div
```

Opcode_Function_Regimm

PLA_OR.txt

Table 3 – Consecutive Control Signal Configuration in PLA_OR.txt

Port name	Classification	Bit	Description
RegDst	Output	2-bit	Register Control signal
RegDatSel	Output	2-bit	Register Write Data Selection signal
RegWrite	Output	1-bit	Register Control signal
SEUmode	Output	1-bit	Extender Control signal
ALUsrcB	Output	2-bit	ALU Input Selection signal
ALUctrl	Output	2-bit	ALU Control signal
ALUop	Output	5-bit	ALU Operation Control signal
DataWidth	Output	3-bit	Memory Data Control signal
MemWrite	Output	1-bit	Memory Control signal
MemtoReg	Output	1-bit	MEM/ALU Selection signal
Branch	Output	3-bit	Branch Address Control signal
Jump	Output	2-bit	Jump Address Control signal

* **The last 5 bits are reserved. Set them as xxxxx.**

PLA_OR.txt

```
xx_xx_x_x_xx_xx_xxxxx_xxx_x_x_xxx_xx_xxxxx  // 0x00 : sll   $d = $t << a
xx_xx_x_x_xx_xx_xxxxx_xxx_x_x_xxx_xx_xxxxx  // 0x01 : srl   $d = $t >> a
xx_xx_x_x_xx_xx_xxxxx_xxx_x_x_xxx_xx_xxxxx  // 0x02 : sra   $d = $t >>> a
xx_xx_x_x_xx_xx_xxxxx_xxx_x_x_xxx_xx_xxxxx  // 0x03 : sllv  $d = $t << $s
xx_xx_x_x_xx_xx_xxxxx_xxx_x_x_xxx_xx_xxxxx  // 0x04 : srlv  $d = $t >> $s
xx_xx_x_x_xx_xx_xxxxx_xxx_x_x_xxx_xx_xxxxx  // 0x05 : srav  $d = $t >>> $s
xx_xx_x_x_xx_xx_xxxxx_xxx_x_x_xxx_xx_xxxxx  // 0x06 : jr    pc = $s
```

RegDst_RegDatSel_RegWrite_SEUmode_ALUsrcB_ALUctrl_ALUop_DataWidth_Memwrite_MemtoReg_Branch_Jump

Control Signals for SCPU.pdf

M_TEXT_SEG.txt

M_TEXT_SEG.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

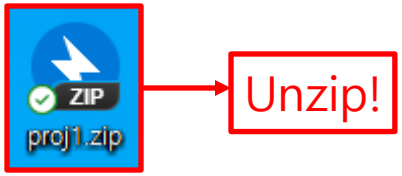
```
00111100_00000010_00010010_00110100
00110100_01000011_01010110_01111000
00111100_00000100_00010001_00100010
00110100_10000101_00110011_01000100

00000000_01100101_00110000_00100011
xxxxxxxx_xxxxxxxx_xxxxxxxx_xxxxxxxx
xxxxxxxx_xxxxxxxx_xxxxxxxx_xxxxxxxx
xxxxxxxx_xxxxxxxx_xxxxxxxx_xxxxxxxx
```

```
//lui $0 $2 0x1234   lui : load upper immediate
//ori $2 $3 0x5678   ori : or immediate
//lui $4 0x1122
//ori $5 $4 0x3344
```

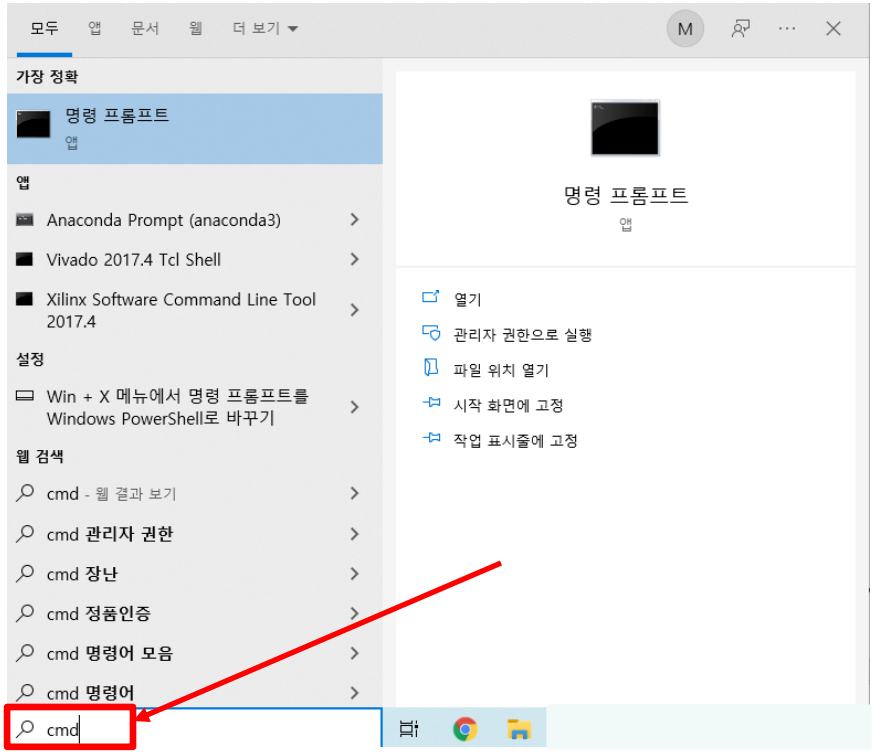
*You can change underbar(_) position for convenience

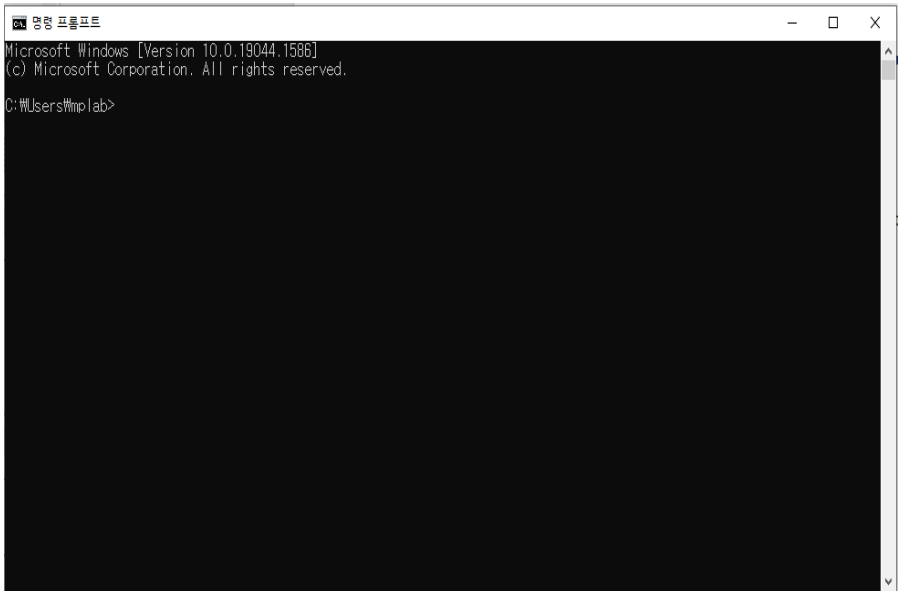
R-Type:	op	$rs	$rt	$rd	shamt	func
I-Type:	op	$rs	$rt	imm16		
J-Type:	op	imm26				

Simulation (1/7)

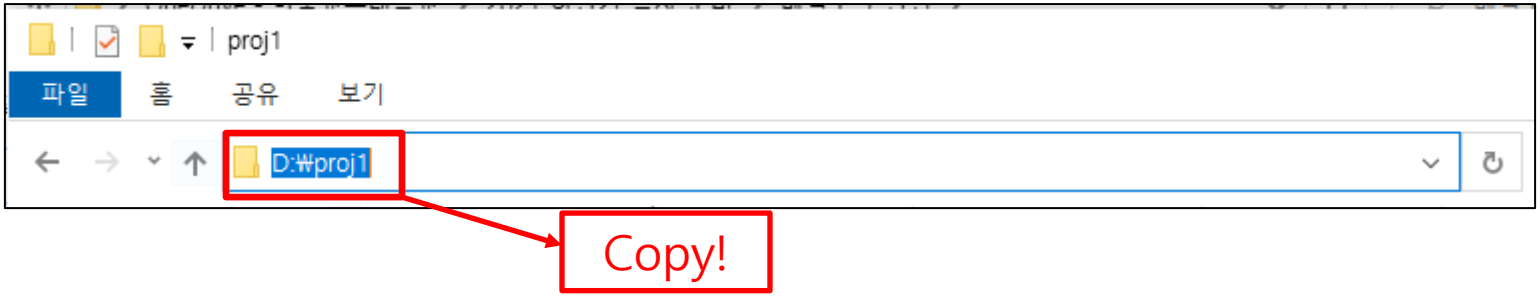


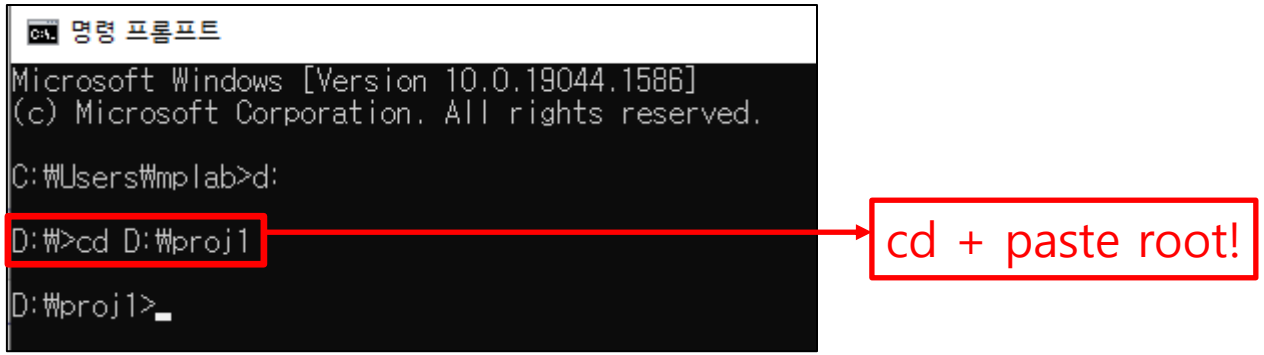
Unzip!

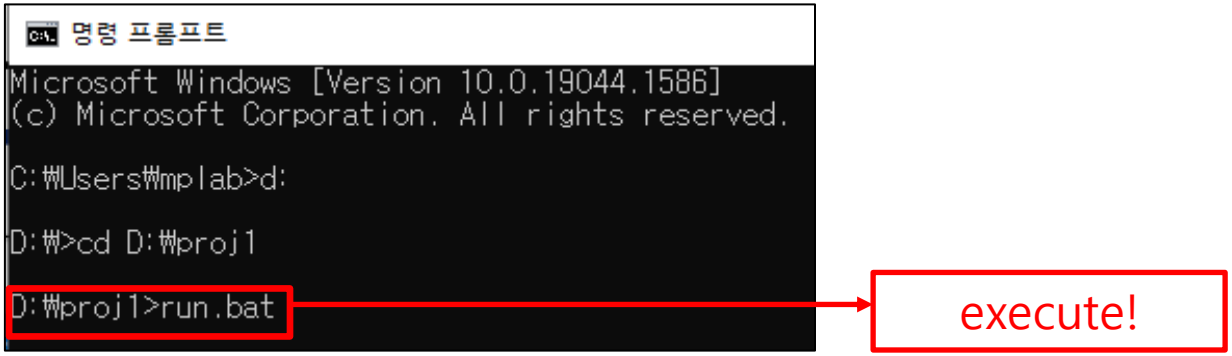




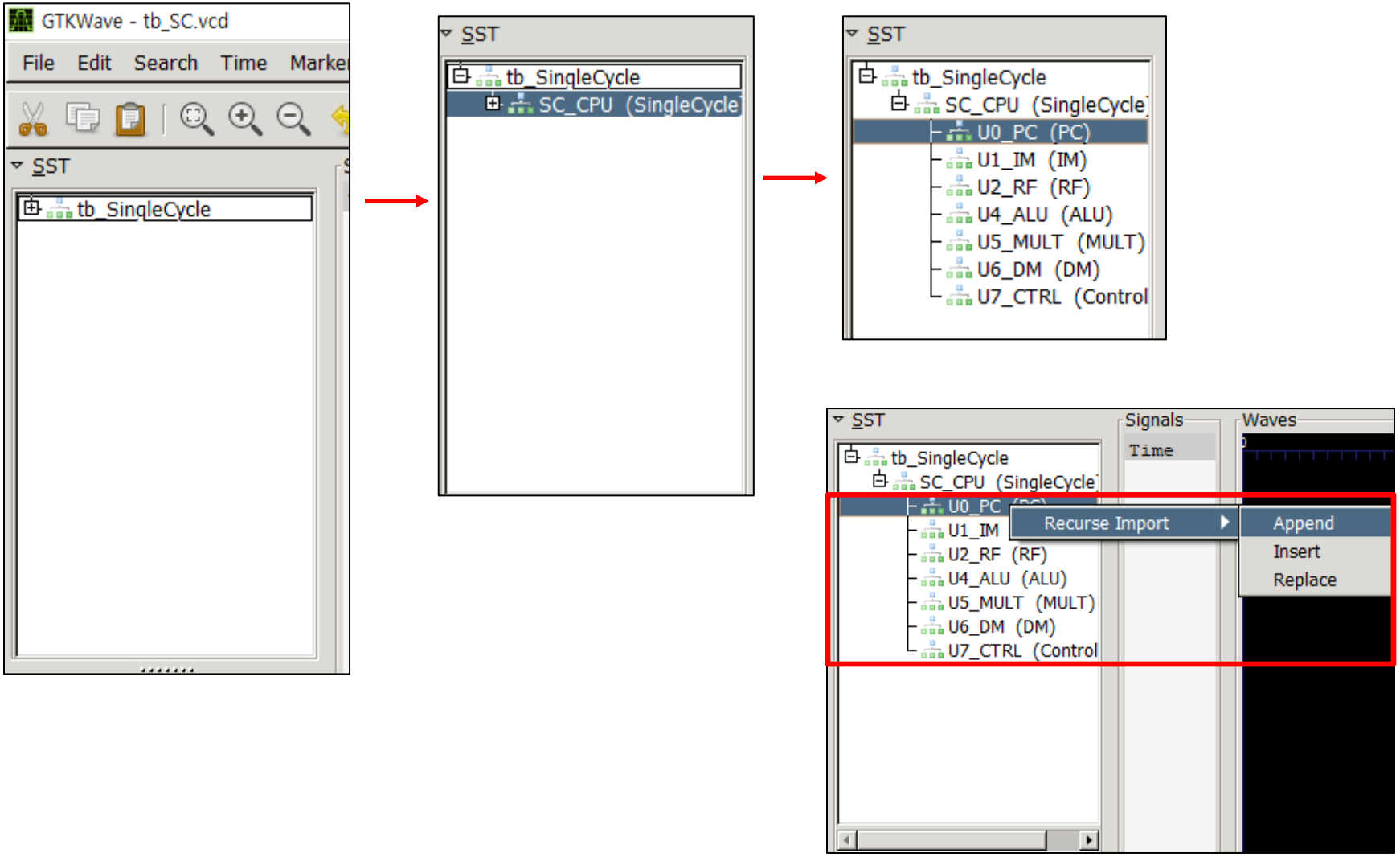
Simulation (2/7)



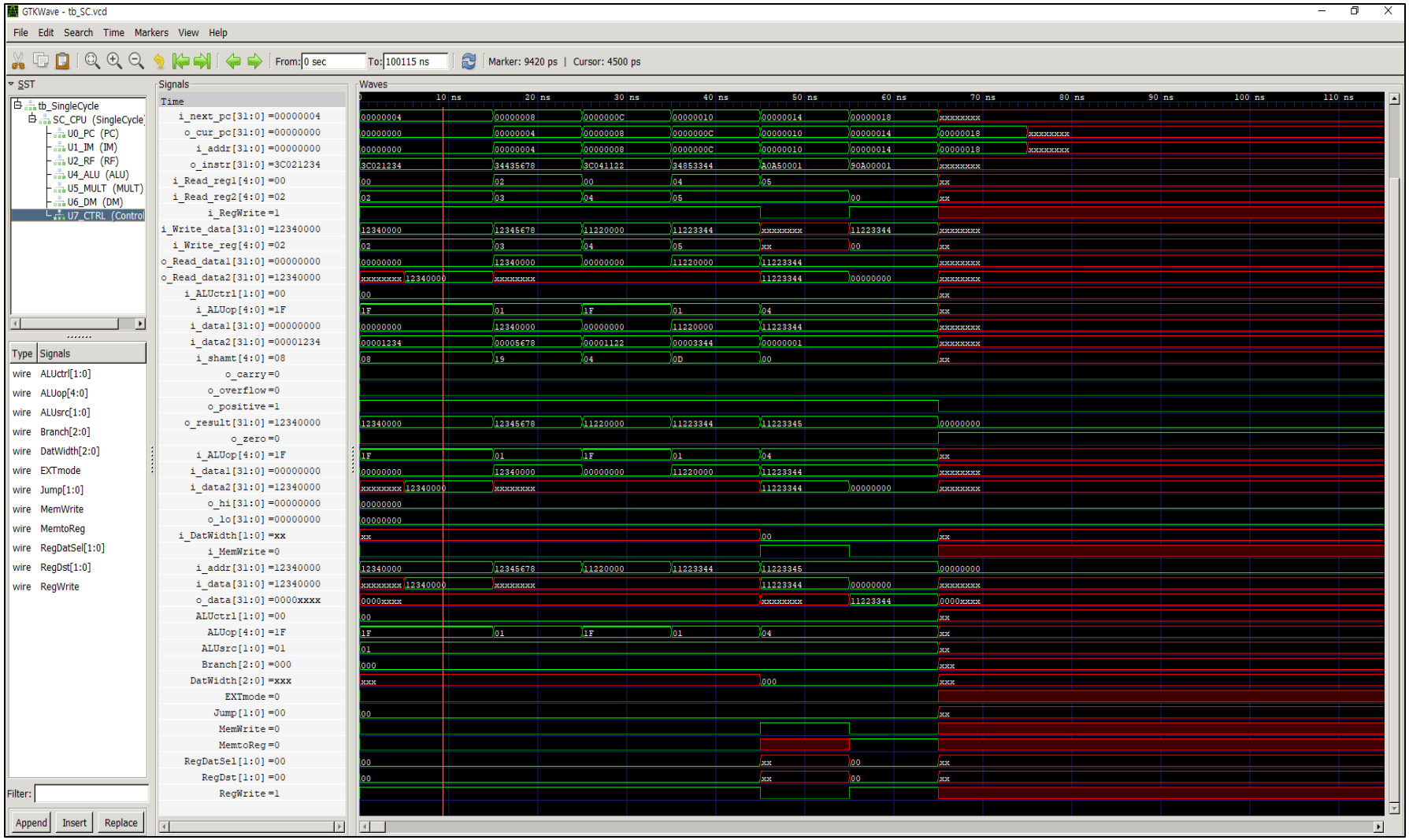




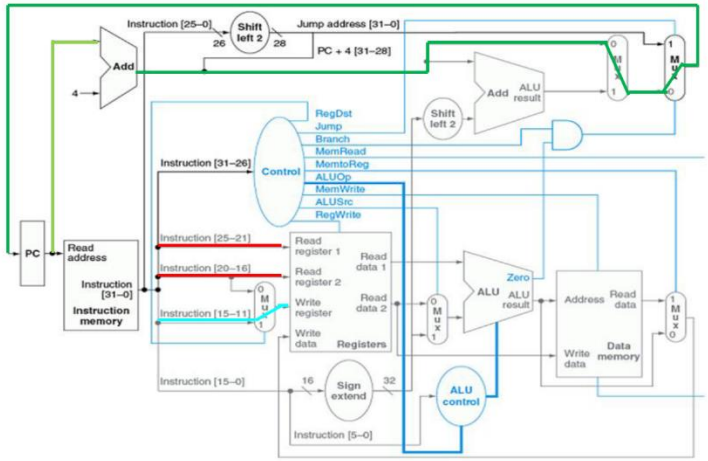
Simulation (3/7)

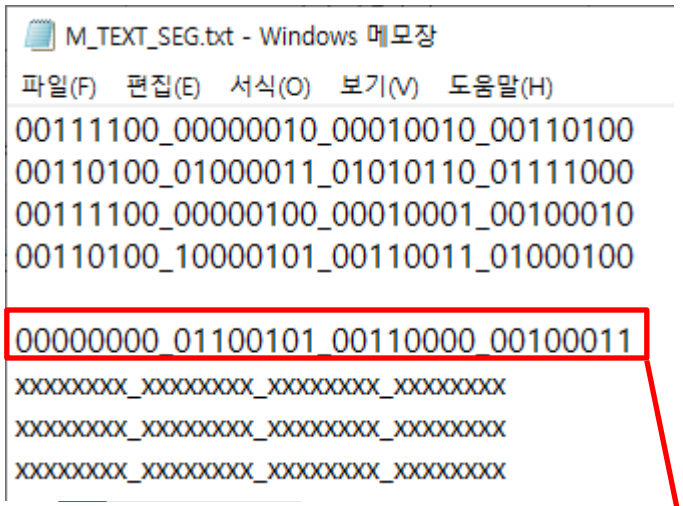


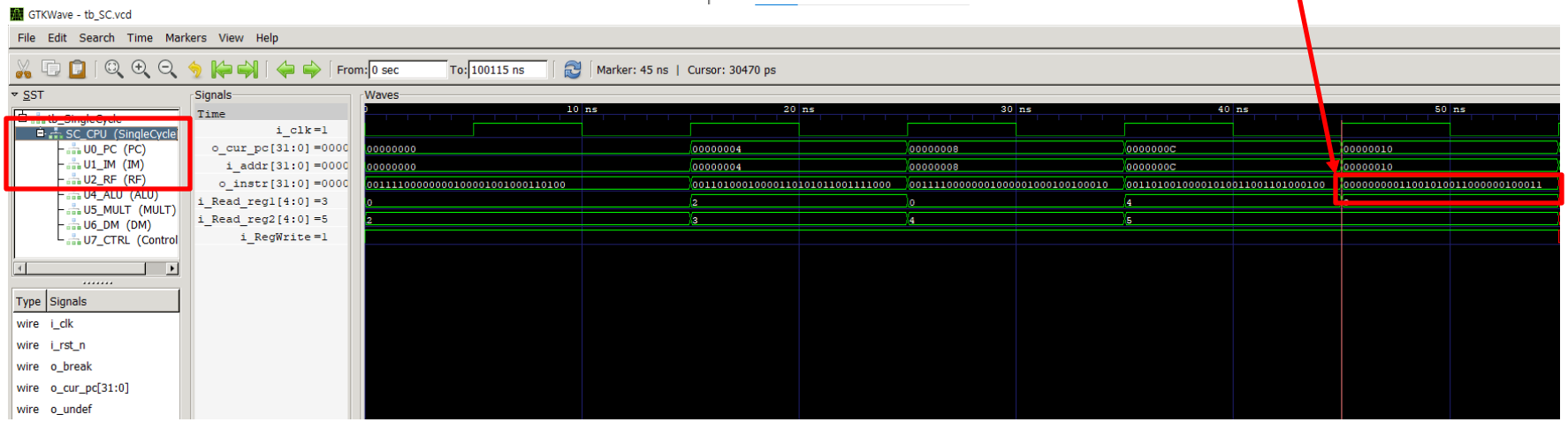
Simulation (4/7)



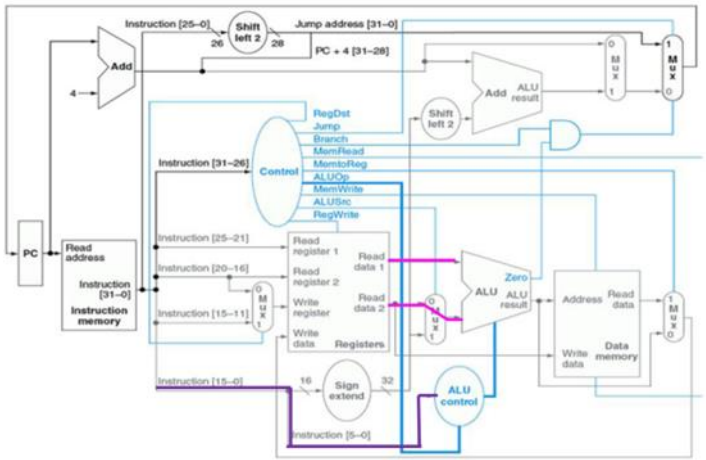
Simulation (5/7)

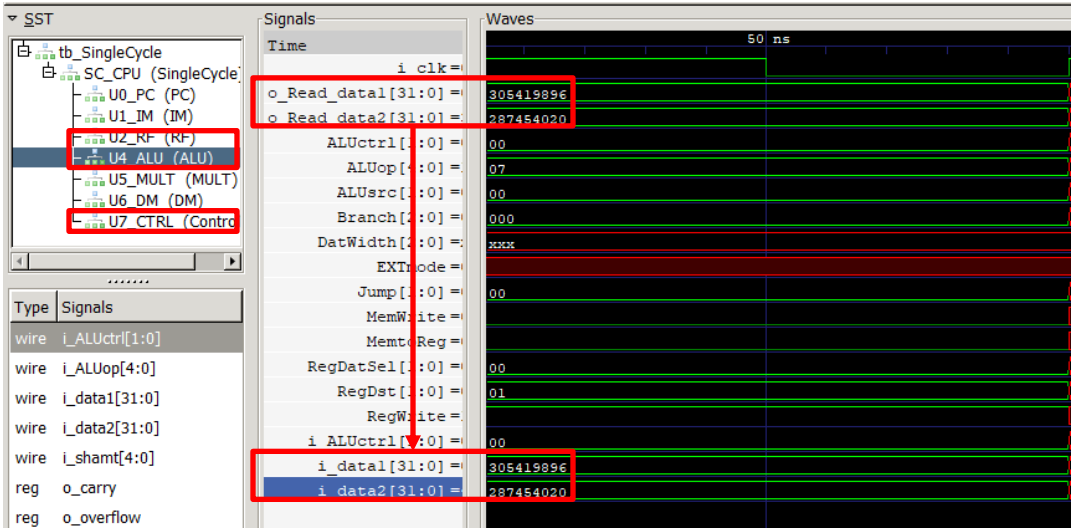




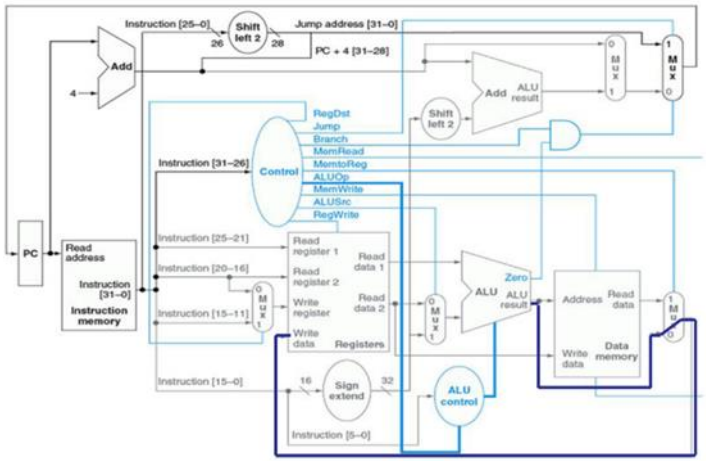


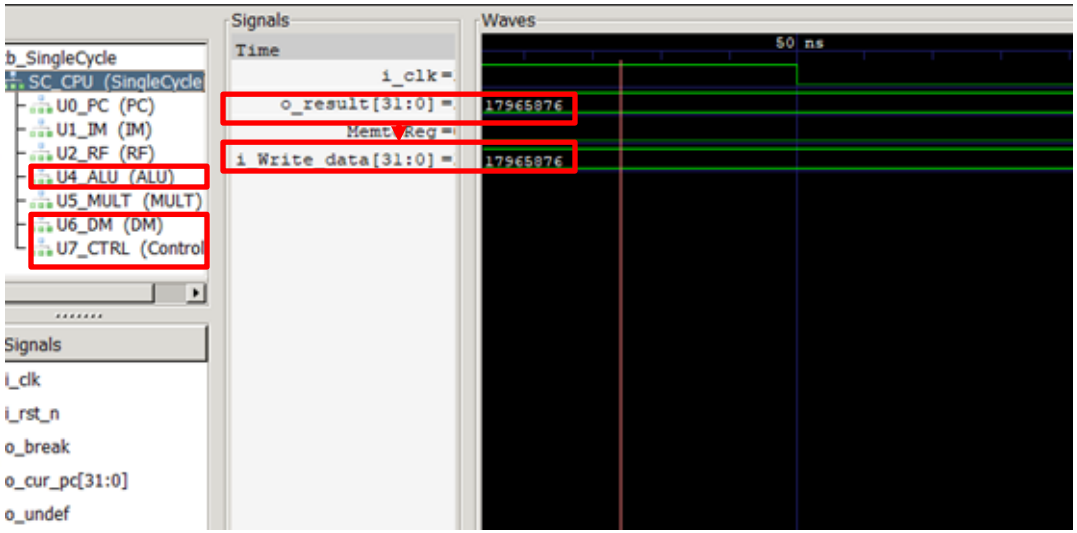
Simulation (6/7)





Simulation (7/7)





Thank You