國立清華大學資訊工程學系

CS 4100 --- 計算機結構

88 學年度上學期期末考

1. (10%) In Fig. 1 on the attached sheet, please mark on the figure the active functional units, asserted control lines, and active data lines, when the single-cycle CPU is executing the instruction: `sw $t1,offset($t2)`.
2. (20%) You are given the microprogram of the multicycle CPU, the microcode controller, and the two dispatch ROMs in Fig 2 on the attached sheet. (a) For each of the circled blocks in the microcode controller, write down on the figure its output values, assuming that the controller is executing the third microinstruction (i.e., `MEM1`) in the microprogram. (b) Suppose we want to merge the two dispatch ROMs into one. Show your modification in the microprogram and the dispatch ROMs.
3. (15%) In Fig. 3 on the back of the attached sheet, a pipelined datapath is shown. (a) Mark on the figure the active data/control lines and functional units in the MEM stage. Suppose the EX/MEM register contains a `beq` instruction and the branch is taken. (b) Suppose the following code segment is executed

   ```
   40  beq $1,$3,7
   44  and $12,$2,$5
   48  or  $13,$2,$6
   52  add $14,$4,$2
   56  slt $15,$6,$7
   ```

   Suppose further that `beq` is taken, i.e., `$1=$3`. According to Fig. 3, how many subsequent instructions after `beq` will be executed before the branch really takes effect? **(Give your reasons to get points!)** (c) Where does `beq` branch to?
4. (10%) Consider the following instruction sequence:

   ```
   add $1,$2,$3
   lw  $1,10($4)
   ```

   If they were executed on the datapath shown in Fig.4 on the back of the attached sheet, will the forwarding logic works correctly? Give you explanation
5. (15%) Suppose we want to execute the following code segment on the pipeline CPU of the text book:

   ```
   add $2,$5,$4
   add $4,$2,$5
   lw  $5,100($2)
   add $3,$2,$5
   ```

   Suppose there is no hardware supports for forwarding and stalling, but the register file can write a register at the first half of a cycle and read it at the second half of the cycle. (a) How many NOPs and where are you going to add to make the code segment execute correctly on our pipeline? (b) Suppose the pipeline stalls, but no forwards, when there is data hazard. How many cycles will the above code segment execute?
6. (15%) Suppose that a computer has a cache of 64KB. Each block contains one 32-bit word. The cache is direct-mapped and write-through. The memory address has 32 bits, and is byte-addressed. (a) How many bits are there in the tag and index fields of the address format? (b) Explain why the CPU does not need to check for write hit when writing to the cache. (c) How many bits are there in the tag and index fields of the address format, if the cache is 8-way set-associative?
7. (5%) Assume an instruction cache miss rate for a program is 2% and data cache miss rate is 4%. Suppose 40% of the instructions executed are loads and stores. If a machine has a CPI of 2 without any memory stalls and the miss penalty is 40 cycles for all misses, determine the CPI of the program with memory stall. (You need to consider both instruction and data misses.)
8. (10%) Consider the code segment in Problem 5 again. Suppose our CPU does not have TLB, but the related page table entries, instructions and data are all in the cache. How many times the CPU have to access the cache in order to execute this code segment? (You need to consider virtual address translation, instruction and data accesses, etc.) **Give your reasons.**

Return this sheet with your answer book! **Name:**_____ **ID:**_____

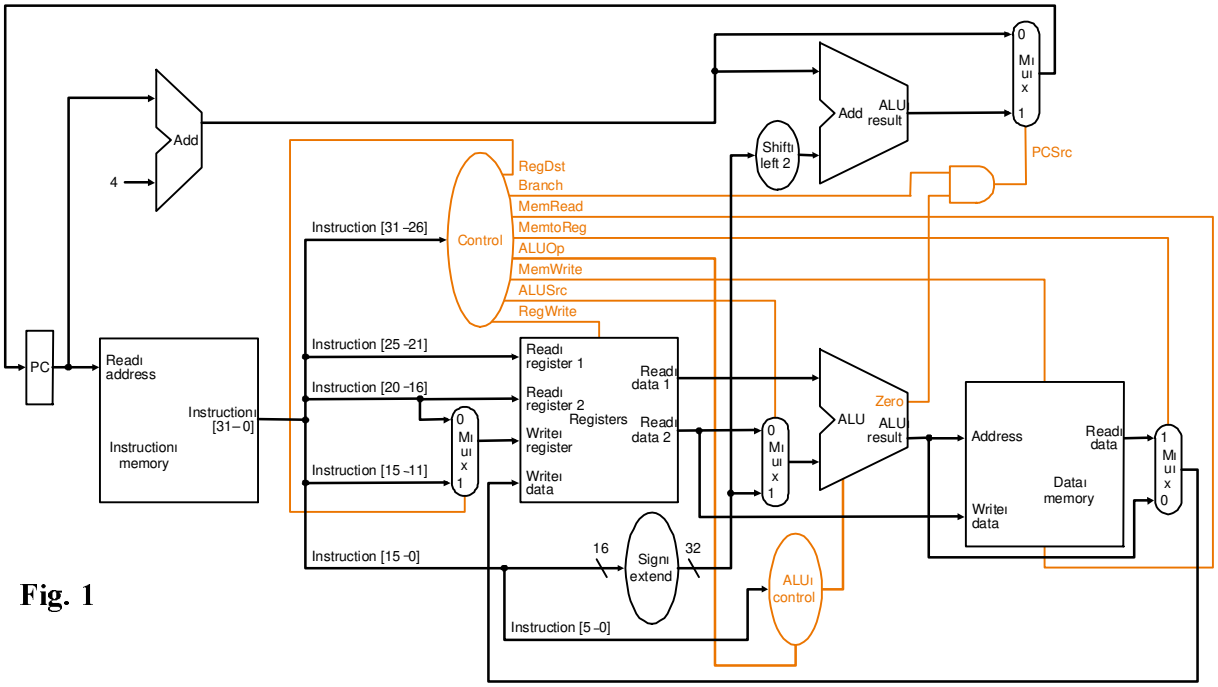
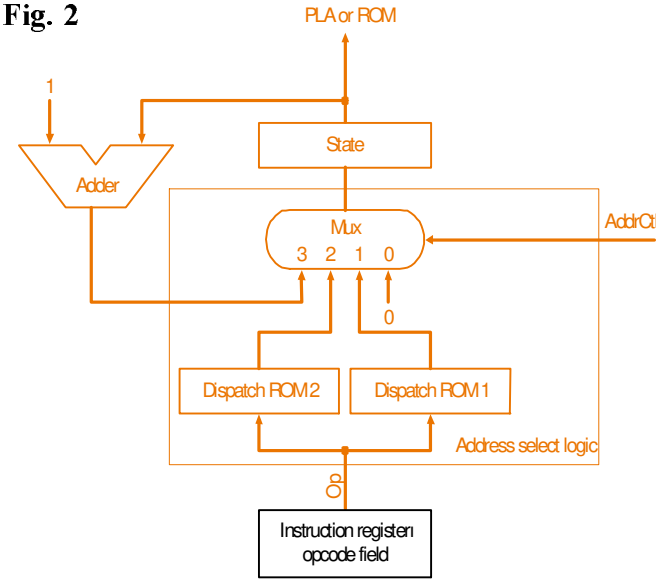


Fig. 1

Label	ALU control	SRC1	SRC2	Register control	Memory	PCWrite control	Sequencing
Fetch	Add	PC	4		Read PC	ALU	Seq
	Add	PC	Extshft	Read			Dispatch 1
Mem1	Add	A	Extend				Dispatch 2
LW2					Read ALU		Seq
				Write MDR			Fetch
SW2					Write ALU		Fetch
Rformat1	Func code	A	B				Seq
				Write ALU			Fetch
BEQ1	Subt	A	B			ALUOut-cond	Fetch
JUMP1						Jump address	Fetch

Fig. 2



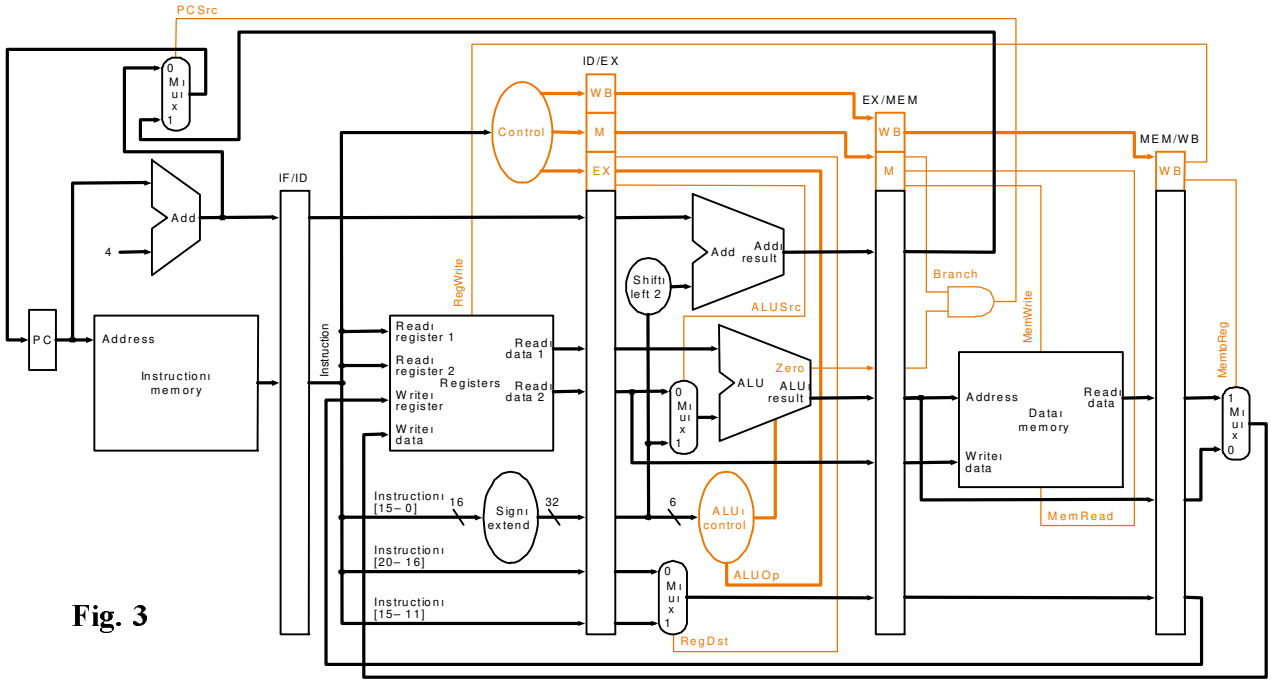

PCSrc
M u x
0
1
Add
4
PC
Address
Instruction memory
IF/ID
Instruction
Read register 1
Read register 2
Registers
Write register
Write data
Read data 1
Read data 2
RegWrite
Instruction [15–0]
Instruction [20–16]
Instruction [15–11]
16
Sign extend
32
ID/EX
Control
WB
M
EX
Add
Add result
Shift left 2
ALUSrc
ALU
Zero
ALU result
6
ALU control
ALUOp
RegDst
EX/MEM
MEM/WB
Branch
MemWrite
MemRead
MemtoReg
Address
Data memory
Read data
Write data

Fig. 3

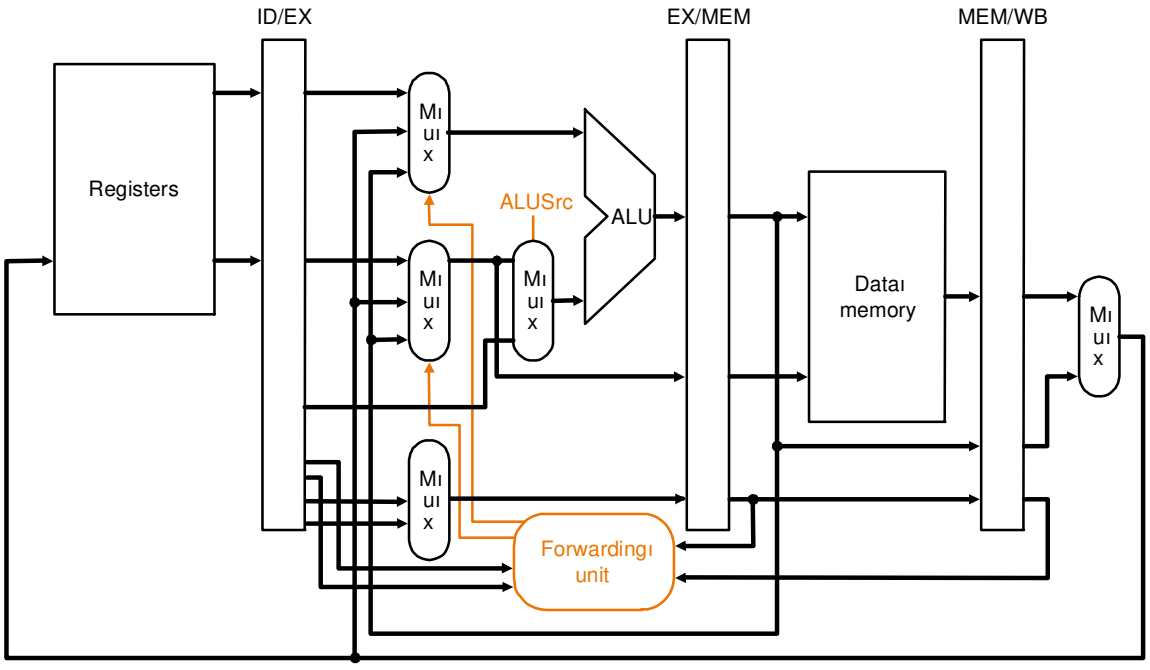

ID/EX
EX/MEM
MEM/WB
Registers
M u x
ALUSrc
ALU
Data memory
Forwarding unit

Fig. 4