题目	一（13）	二（8）	三（13）	四（6）	五（60）	总分
得分						
批阅人						

自觉遵守考场纪律 如考试作弊 此答卷无效

姓名

学号

一、Translate into Chinese (total 13 points)

1. **(4 points)** Processors make use of instruction pipelining to speed up execution. Pipelining involves breaking up the instruction cycle into a number of separate stages that occur in sequence. Instructions move through these stages, as on an assembly line, so that each stage can be working on a different instruction at the same time.

2. **(3 points)** A superscalar processor typically fetches multiple instructions at a time and then attempts to find nearby instructions that are independent of one another and can therefore be executed in parallel.

3. **(3 points)** The control signals generated by the control unit cause the opening and closing of logic gates, resulting in the transfer of data to and from registers and the operation of the ALU.

4. **(3 points)** Cache memory is in intended to give memory speed approaching that of the fastest memories available, and at the same time provide a large memory size at the price of less expensive types of semiconductor memories.

二、**Fill blanks (total 8 points)**

1. **(2 points)** RISC is abbreviation of _____.

2. **(2 points)** CISC is abbreviation of _____.

3. **(1 point)** AB+CD-* of the corresponding infix expression is _____.

4. **(1 point)** A-B-C-D of the corresponding reverse Polish expression is _____.

5. **(2 points)** There are 155 registers in a computer, 11 of which are left for global use and the others constitute a number of register windows. Each window is consisted of 24 register (8 parameter registers, 8 local registers, and 8 temporary registers). The register window number is _____, and the register window structure supports procedures with depth of _____ at most without window overlap.

三、**Fill blanks according to given contents (total 13 points)**

1. **(10 points)** In computer architecture, a branch predictor is a digital circuit that tries to guess which way a branch (e.g. an if-then-else structure) will go before this is known for sure. The purpose of the branch predictor is to improve the flow in the instruction pipeline. Branch predictors play a critical role in achieving high effective performance in many modern pipelined microprocessor architectures.

Two-way branching is usually implemented with a conditional jump instruction. A conditional jump can either be "not taken" and continue execution with the first branch of code which follows immediately after the conditional jump - or it can be "taken" and jump to a different place in program memory where the second branch of code is stored.

It is not known for certain whether a conditional jump will be taken or not taken until the condition has been calculated and the conditional jump has passed the execution stage in the instruction pipeline.

Without branch prediction, the processor would have to wait until the conditional jump instruction has passed the execute stage before the next instruction can enter the fetch stage in the pipeline. The branch predictor attempts to avoid this waste of time by trying to guess whether the conditional jump is most likely to be taken or not taken. The branch that is guessed to be the most likely is then fetched and speculatively executed. If it is later detected that the guess was wrong then the speculatively executed or partially executed instructions are discarded and the pipeline starts over with the correct branch, incurring a delay.

Fill the following blanks according to the above description

(1) **(2 points)**Give a title for the above phases: __Branch predictor_____

(2) **(2 points)**Branch predictors are critical for the instruction of _pipeline_____

(3) **(2 points)**A conditional jump is _"taken"_ if it turns to a different place in program memory that does not follow immediately after the conditional jump.

(4) **(2 points)**The “taken” or “not taken” is decided after the stage of _execution in the instruction pipeline has passed.

(5) **(2 points)**If the prediction is _wrong_ then the instructions fetched and partially executed instructions following the conditional jump are discarded and the pipeline starts over with the correct branch.

2. **(3 points)** An alternative to a hardwired control unit is a microprogrammed control unit, in which the logic of the control unit is specified by a microprogram which consists of a sequence of instructions in a microprogramming language. These are very simple instructions that specify micro-operations.

A microprogrammed control unit is a relatively simple logic circuit that is capable of sequencing through microinstructions and generating control signals to execute each microinstruction.

As in a hardwired control unit, the control signals generated by a microinstruction are used to cause register transfer and ALU operations

Fill the following blanks according to the above description

(1) **(1 point)** Give a title for the above phases: _Microprogrammed control unit_

(2) **(1 point)** In the microprogrammed control unit, the microprogram is used to specify _the logic of the control unit_

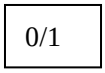
(3) **(1 point)** In the microprogrammed control unit, micro-operations are specified by _microinstructions_

四、Review Questions (total 6 points)

1. **(3 points)** Various techniques can be used to predict whether a branch will be taken or not. Dynamic branch strategies attempt to improve the accuracy of prediction by recording the history of conditional branch instructions in a program. For example, one or more bits can be associated with each conditional branch instruction that reflect the recent history of the instruction. With a single bit, all that can be recorded is whether the last execution of this instruction resulted in a branch or not. A shortcoming of using a single bit appears in the case of a conditional branch instruction that is almost always taken, such as a loop instruction. With one bit of history, an error in prediction will occur twice for each use of the loop: one entering the

loop, and once on exiting. Now the following is a loop, please give the reason why two errors will occur for only
one history bit is used.

```
        LD    R3,0
        LD    R4,1
Again: SUB    R3, R3, R4
        BEQ   R4, 100, Exit      ; Branch if R4=100, this is a conditional branch instruction.
        ADD   R4,R4,1
        JMP   Again
```



The one history bit is shown as the figure. It is set (1) or reset (0).

2. **(3 points)** An instruction is being executed, at the moment an interrupt occurs, and an interrupt request is sent to CPU. Please answer how the CPU responds to the interrupt request?

五、Problems

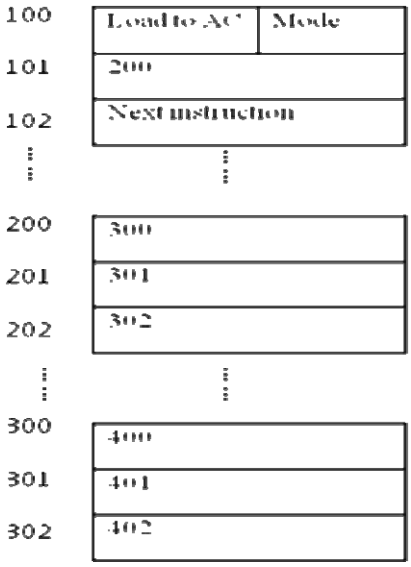
1. **(5 points)** Consider a memory system that uses a 32-bit address to address at the byte level, plus a cache that uses a 32-byte line size. Assume a 2-way set associative cache with a tag field in the address of 8 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in set, number of sets in cache, number of lines in cache.

2. （**3 points**）Consider a memory with contents appearing in the following figure, in which the instruction staring at location 100 includes an opcode field, an addressing mode field, and an addressing field. Determine the operand to be loaded for the following address modes:

(1) (1 point) Immediate

(2) (1 point) Direct

(3) (1 point) Indirect



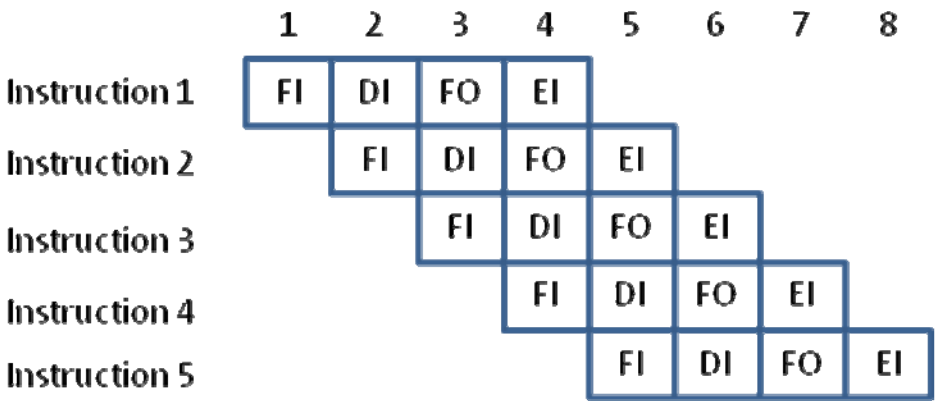
3. **(8 points)** Consider an instruction pipeline of four-stages, fetch instruction (FI), decode instruction (DI), fetch operands (FO), and execute instruction (EI).

(1) Draw a diagram to show the timing of the instruction pipeline operation for 5 successive instructions without branch;

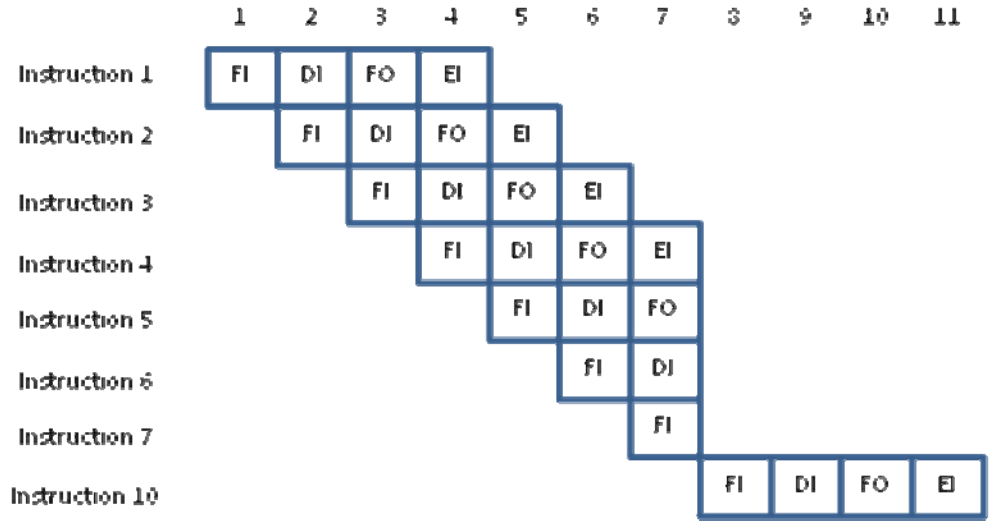
(2) Draw a diagram to show the timing of the instruction pipeline operation for the case of a branch is taken in the 4rd instruction and jumps to instruction 10;

(3) Calculate the speed up factors of the above two cases using pipeline.

Answer (1)



Answer (2)



Answer (3)
speed up factor (1): 4*5/8=2.5
speed up factor (2): 4*5/11=20/11

4. **(6 points)** Consider the following program

```
        LD    R1, 100
        LD    R2, 1
LP:     ADD   R1, R1, R2
        BEQ   R2, 100, EXIT
        ADD   R2, R2, 1
        JMP   LP
```

A compiler for a RISC machine will introduce delay slots into this code so that the processor can employ the delayed branch mechanism.

(1) Show the code using delayed branch method to implement the loop.

(2) In the code you can insert an NOOP instruction after the BEQ instruction to deal with the conditional branch or you do not use the NOOP, but the two ways produce different results in register R2 after the loop. Show the result of R2 after the loop.

Answer: (1)

```
        LD    R1, 100
        LD    R2, 1
LP      ADD   R1, R1, R2
LP1     BEQ R2. 100, EXIT
        NOOP
        ADD R2, R2, 1
        JMP LP1
```

```
        ADD   R1, R1, R2
(2) R2 = 100

Or   (1)
        LD    R1, 100
        LD    R2, 1
   LP   ADD   R1, R1, R2
   LP1  BEQ R2. 100, EXIT
        ADD R2, R2, 1
        JMP LP1
        ADD   R1, R1, R2
(2) R2 = 101
```

5. **(10 points)** There are some dependencies in the following code fragment.
   (1) Point out 4 dependencies of which at least,
   (2) increase instruction parallelism using register renaming, and
   (3) Point out dependencies after register renaming.

```
I1: R1 ← R3
I2: R3 ← R3 ADD R5
I3: R4 ← R3 + 1
I4: R3 ← R5 + 1
I5: R6 ← R3 SUB R4
```

Answer: (1)
I1, I2: Read-Write dependence (Anti-dependency)
I1, I4: Read-Write dependence (Anti-dependency)
I2, I3: Write-Read dependency (True data dependency)
I2, I4: Read-Write dependence (Anti-dependency), Write-Write dependency (Output dependency)
I3, I4: Read-Write dependence (Anti-dependency)
I3, I5: Write-Read dependency (True data dependency)
I4, I5: Write-Read dependency(True data dependency)

(2)

```
I1: R1a ← R3a
I2: R3b ← R3a ADD R5a
I3: R4b ← R3b + 1
I4: R3c ← R5a + 1
I5: R6a ← R3c SBB R4b
```

(3)
I2, I3: Write-Read dependency (True data dependency)

I3, I5: Write-Read dependency (True data dependency)
I4, I5: Write-Read dependency(True data dependency)

6. **(10 points)** There for instructions as I1~I6, it is assumed that:
I1 requires two cycles to execute.
I3 and I4 conflict for the same functional unit.
I5 depends on the value produced by I4.
I5 and I6 conflict for a functional unit.
Instructions are fetched two at a time and passed to the decode unit. Now please complete the process based on the clock cycles according to Out- of-order Issue with Out-of-Order Completion. The figure has been given. The window size is assumed to be 8, that is, it can hold 8 instructions.

Decode		Window	Execute Used by I1	Execute Used by I2,I5,I6	Execute Used byI3,I4	Write		Cycle
								1
								2
								3
								4
								5
								6

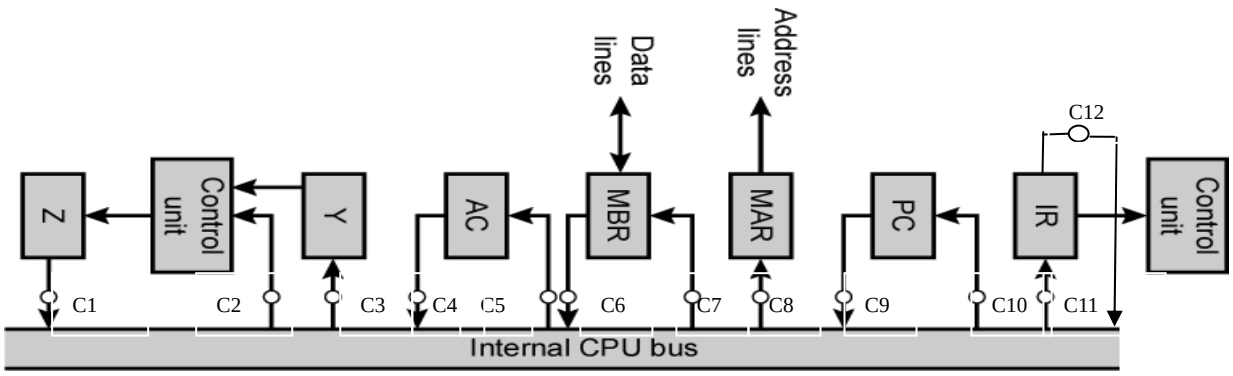


Figure 1. Internal CPU and Control Signals for a CPU

7. **(10 Points)** Show the micro-operations and control signals needed in the following table for the CPU in Figure 1. to fetch and execute the instruction of ADD a number to the Accumulator (the mnemonic is **ADD** AC) with 2 different addressing mode:

(1) Direct address; (2) Indirect address.

(**Note:** the cycles in Figure 1 represent gates and C with subscript is the control signals to the corresponding gates.)

Solution:

Table

Instruction sub-cycles	Micro-operations	Active control signals
Fetch: **(4 points)**		
Case 1 Execute (direct) : **(2 points)**		

Case2	Indirect addressing: **(4 points)**		
	Execute:	(Note: Do Not Write)	(Do Not Write)

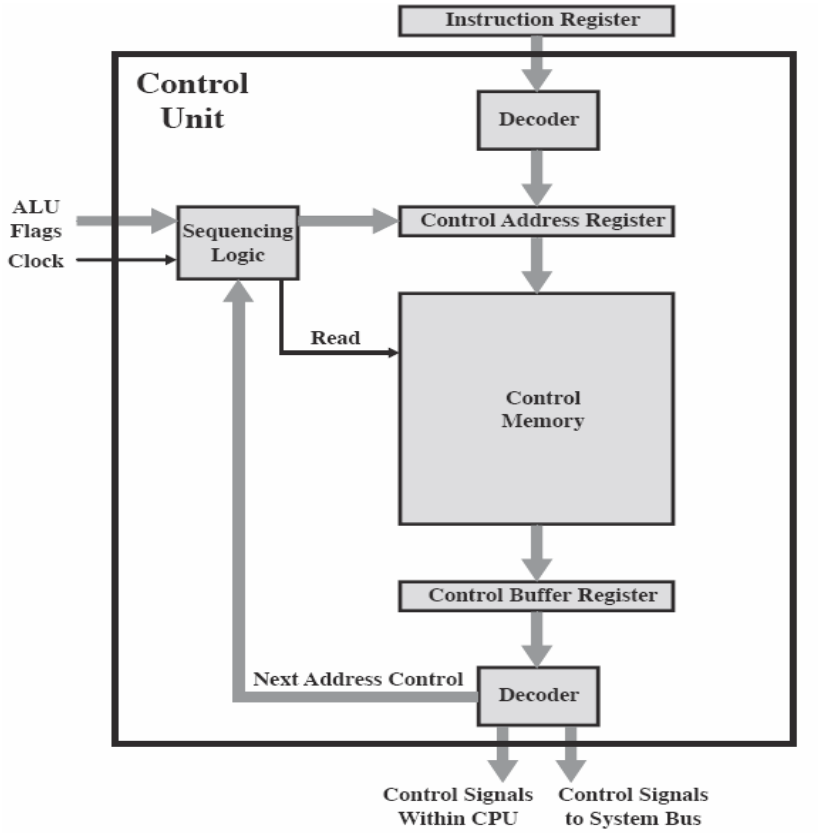


Figure 2. functioning of micro-programmed control unit

8. **(8 Points)** Figure 2. presents the diagram of functioning of micro-programmed control unit. Please try to answer the following questions:

(1) (2 point) What is the purpose of the control memory?

(2) (2 points) How to get the next micro-instruction address?

(3) (4 points) What is the progress of the control unit to function?