东南大学考试卷（ A 卷）

课程名称 计算结构（英文） 考试学期 16-17-2 得分

适用专业 信息学院 考试形式 闭卷 考试时间长度 120 分钟

题目	一（12）	二（10）	三（10）	四（10）	五（10）	六（48）	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

线 封 密

姓名

学号

一、Translate into Chinese(Total 12 points)

1. **(3points)** Processors make use of instruction pipelining to speed up execution. In essence, pipelining involves breaking up the instruction cycle into a number of separate stages that occur in sequence. The occurrence of branches and dependencies between instructions complicates the design and use of pipelines.

   处理器使用指令流水线来加速指令执行的过程。本质上，流水线将指令分成几个分离的按顺序进行的阶段。指令转移和指令间的相关性使流水线的设计和使用变得复杂。

2. **(2points)** Delayed branch, a way of increasing the efficiency of the pipeline, makes use of a branch that does not take effect until after execution of the following instruction (hence the term delayed).

   延迟转移是一种提高流水线效率的方法，它使转移指令直到下一条指令执行之后才发生作用（因此称为延迟）。

3. **(4points)** A superscalar processor is one in which multiple independent instruction pipelines are used. Each pipeline consists of multiple stages, so that each pipeline can handle multiple instructions at a time. Multiple pipelines introduce a new level of parallelism, enabling multiple streams of instructions to be processed at a time.

   超标量处理器中使用多个相互独立的指令流水线。每条流水线由多个阶段组成，因此每条流水线可以在同一时刻处理多条指令。多条流水线引入了一种新层次的并行性，使得多条指令流可以在同一时刻被处理。

4. **(3points)** One technique for implementing a control unit is referred to as hardwired implementation, in which the control unit is a combinatorial circuit. Its input logic signals, governed by the current machine instruction, are transferred into a set of output control signals.

   被称为硬连线实现的技术是一种实现控制单元的技术，该实现方式的控制单元是一个组合电路。由当前机器指令控制的输入逻辑信号被转换成一系列输出控制信号。

二、Fill blanks (Total 10points, 2 points/blank)

1. A computer has a cache of 8k Bytes and a main memory of 16M Bytes dressed on Bytes. The address format is 13, 8, and 3 bits for Tag, Set, and Word respectively. It is needed to compare tags to search a word in the cache. For one word access, the times of tag comparison is (4) at most.
2. Computer memory is organized into a hierarchy. At the highest level (closest to the processor) are the processor (registers).
3. In (immediate) addressing, the operand value is present in the instruction.
4. In register windows the (temporary registers) at one level are physically the same as the parameter registers at the next lower level.
5. Some dependencies, such as the write-write dependency of a symbol, can be solved by (register renaming).

三、Select A, B, C, or D corresponding to the best answer from the given items (Total 10 points, 2 points/blank)

1. When a processor accesses a cache, the reference address is (C)
   A. Register address　　B. Cache address
   C. Main memory address　　D. Disk address
2. By using the (D), a word is retrieved based on a portion of its contents rather than its address.
   A. direct access　　B. random access
   C. sequential access　　D. associative access
3. If most memory references are relatively near to the instruction being executed, then the use of (D) addressing saves address bits in the instruction.
   A. direct　　B. indirect
   C. register　　D. relative

4. To perform a sequence of 16 instructions using pipeline, each instruction is divided into 4 stages FI, DI, EI and WO, where each of the stages require **1 cycle**, **1 cycle**, **2 cycles** and **1 cycle** respectively. No branch is in the sequence. The speed up factor of the pipeline is (B)

   A. 5 B. 16/7 C. 4 D. 64/19

5. The degree of instruction-level parallelism is determined by the frequency of (C) and procedure dependency in the code.

   A. Read-Write dependency B. Write-Write dependency

   C. Write-Read dependency D. Resource conflict

四、**Answer according to given contents (Total 10 points,2 points/blank)**

Based on the current microinstruction, condition flags, and the contents of the instruction register, a control memory address must be generated for the next microinstruction. A wide variety of techniques have been used. We can group them into three general categories. These categories are based on the format of the address information in the microinstruction: two address fields, single address field, and variable format.

The simples approach is to provide two address fields in each microinstruction. A multiplexer is provided that serves as a destination for both address fields plus the instruction register. Based on an address-selection input, the multiplexer transmits either the opcode or one of the two addresses to the control address register (CAR). The CAR is subsequently decoded to produce the next microinstruction address. The address-selection signals are provided by a branch logic module whose input consists of control unit flags plus bits from the control portion of the microinstruction.

Although the two-address approach is simple, it requires more bits in the microinstruction than other approaches. With some addition logic, saving can be achieved. A common approach is to have a single field. With this approach, the options for next address are address field, instruction register code, and next sequential address.

The address-selection signals determine which option is selected. This approach reduces the number of address fields to one. Note, however, that the address field often will not be used. Thus, there is some inefficiency in the microinstruction coding scheme.

Another approach is to provide for two entirely different microinstruction formats. One bit designates which format is being used. In one format, the remaining bits are used to activate control signals. In the other format, some bits drive the branch logic module, and the remaining bits provide the address. With the first format. The next address is either the next sequential address or an address derived from the instruction register. With the second format, either a conditional or unconditional branch is being specified. One disadvantage of this approach is that one entire cycle is consumed with each branch microinstruction. With the other approaches. Address generation occurs as part of the same cycle as control signal generation, minimizing control memory accesses.

The approaches just described are general. Specific implementations will often involve a variation or combination of these techniques.

(1) Give a title for the above phases:

Microinstruction address format

(2) What are the “other approaches” underlined in the third phase?

One address approach and variable address approach

(3) What are the “these techniques” underlined in the last phase?

Two address approach, one address approach and variable address approach

(4) What are the “two entirely different microinstruction formats” underlined in the fifth phase?

In one format, the remaining bits are used to activate control signals. In the other format, some bits drive the branch logic module, and the remaining bits provide the address. (with or without address)

(5) Which of the mentioned address designate techniques has the highest bit efficiency on average? Why?

Variable address approach
Addresses in the address field will often not be used for two address approach and one address approach. Variable address approach makes the bits of the address field be used for control signals in part time.

五、**Questions (Total 10 points)**

1. **(4points)** What is the relationship among direct mapping, associative mapping, and set associative mapping in cache design?

   Set associative mapping is general and include the other two mapping methods, if the set size is a line, it is direct mapping, and if a set include all the lines of the cache, it is associative mapping.

2. **(3points)** What are characteristics of RISC?
   Using a large number of registers;
   Limited and simple instruction sets;
   Optimized pipeline.
3. **(3points)** What is the purpose of an instruction window?

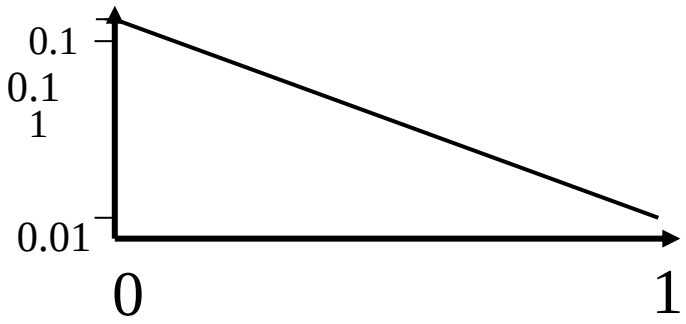
   For an out-of-order issue policy, the instruction window is a buffer that holds decoded instructions. These may be issued from the instruction window in the most convenient order.

六、Problems (Total 48 points)

1. **(5 points)** Suppose that a processor has access to two levels of memory. Level 1 contains 1000 words and has an access time of 0.01 us; level 2 contains 100,000 words and has an access time of 0.1 us. Assume that if a word to be accessed is in level 1, then the processor accesses it directly. If it is in level 2, then the word is first transferred to level 1 and then accessed by the processor. For simplicity, we ignore the time required for the processor to determine whether the word is in level 1 or level 2. Shows the average access time to the two-level memory as a function of the hit ratio H **using a figure**, where H is defined as the fraction of all memory accesses that are found in the faster memory.

Answer:

T=0.01H+(0.1+0.01)(1-H)



2. **(5 points)** An encoded microinstruction format is to be used. Show how an 8-bit micro-operation field can be divided into subfields to specify 38 different actions.

Answer:

5 bits 3 bits

31 actions 7 actiopns

(答案不唯一，总的 actions 满足即可)

3. **(5 points)** To compute Y=(A-B)÷(C+D×E), we can use one-, two-, or three-address instructions, and we can also use zero addresses for some instructions. Zero-address instructions are applicable to a special memory organization, called a stack.

a) Convert Y=(A-B)÷(C+D×E) from infix to reverse Polish;

b) List the instructions using a stack to compute Y=(A-B)÷(C+D×E).

Answer:

a) AB-CDE×+÷

b)

```
PUSH A
PUSH B
SUB
PUSH C
PUSH D
PUSH E
MUL
ADD
DIV
POP Y
```

4. **(5 points)** An instruction is 4 bytes long, and the start address in byte of the instruction is 360000 in decimal. PC-relative addressing mode is used in the instruction to access

an operand and the displacement value in the address field is -30. Determine the address of the target operand.

Answer:

(PC) = 360000+4 =360004
EA = (PC)-30
=359974

5. **(10 points)** A four-way set-associative cache has lines of 16 bytes and a total size of 8kbytes.The main memory of 16-Mbyte is byte addressable.

   a) Show the format of main memory addresses;

   b) CPU is to read a data with address ABCDEF from the Cache-Memory structure. Describe the reading process step by step considering the cases of Hit and Miss;

   c) Draw a block diagram of this cache showing its organization and how the different address fields are used to determine a cache hit/miss.

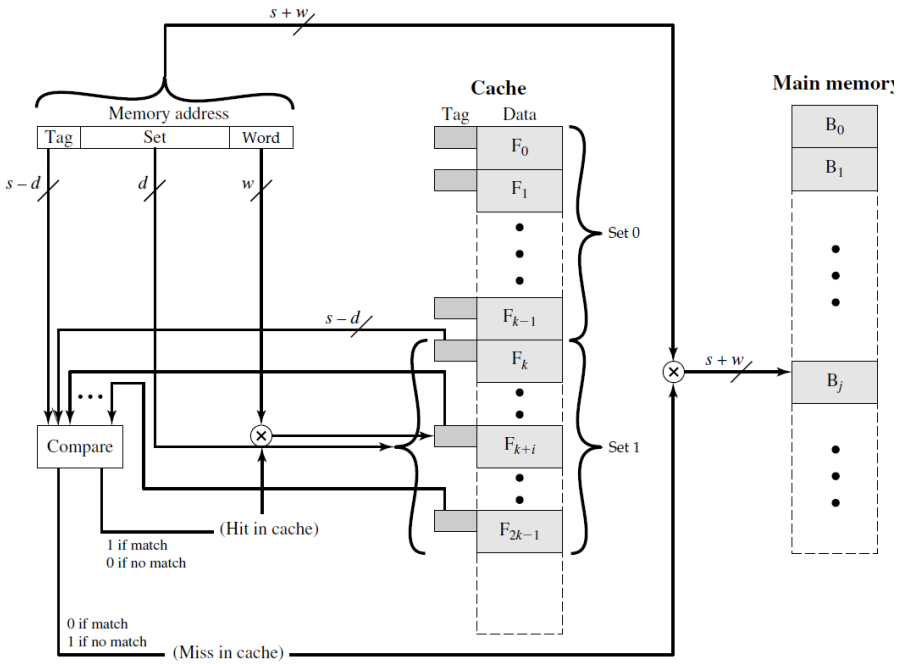
Answer:

a) Address format: 13/7/4

b) For address ABCDEF

tag bits: 1010 1011 1100 1
set bits: 101 1110
word bits:1111
Find set 101 1110 in the cache
Compare the tag 1010 1011 1100 1 with the 4 tags of the set
If Hit, select the line with the tag 1010 1011 1100 1, and select the Byte with word bit 1111
If Miss, search the memory, select the Block including addresses ABCDE0~ABCDEF, put the block onto one line of the set 101 1110 of the cache
Read the Byte of address ABCDEF.

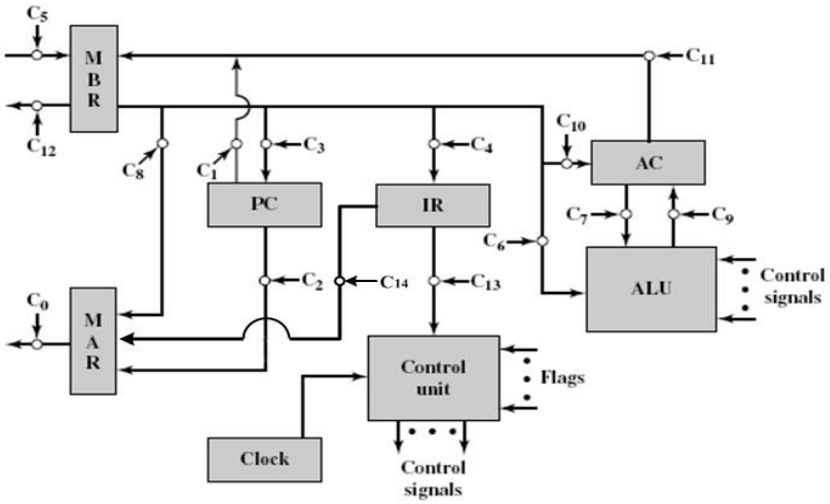
c) Diagram



6. **(10 points)** The instruction "AND X, (Y)" includes the following stages:

   a. Instruction Fetch
   b. Load AC the Data with address X
   c. Indirect Addressing
   d. AND execution
   e. Store AC to memory with address X

   Show the micro-operations of these stages and the corresponding control signals gate in the figure.



a. Instruction Fetch

t1: MAR←(PC) (C_2) ;

t2: MBR←(Memory) (C_0 , C_5 , C_R),

PC←(PC)+1;

t3: IR←(MBR) (C_4) ;

b. Load X to AC

t1: MAR←(IR(address1)) (C_{14});

t2: MBR←(Memory) (C_0 , C_5 , C_R);

t3: AC←(MBR) (C_{10}) ;

c. Indirect addressing

t1: MAR←(IR(address 2)) (C_{14});

t2: MBR←(Memory) (C_0 , C_5 , C_R);

t3: IR(address 2)←(MBR) (C_4); 或者 MAR←(MBR) (C_8)

d. AND (Y) to AC

t1: (MAR←IR(address2) (C_{14}) ;) 对应 c，有或无此步骤

t2: MBR←Memory (C_0 , C_5 , C_R);

t3: AC←(AC) AND (MBR) (C_6, C_7, C_9, C_{ALU});

e. Store AC to X

t1: MAR←IR(address1) (C_{14});

t2: MBR←AC (C_{11});

t3: Memory←(MBR) (C_0 , C_{12}, C_w) .

7. **(8 points)** Consider the following program

Address	Instruction
100	LOAD R1←X
101	ADD R1←R1+1
102	JUMP 105
103	ADD R2←R2+R1
104	SUB R2←R2-R3
105	STORE Z←R1

A compiler for a RISC machine can either insert NOOP instructions or use delayed branch to deal with data dependencies and procedure dependences among instructions in the operation of pipelines. Assume that each instruction is divided into three stages: Fetch, Execute, (with or without) Write,

a. Show the compiled code using NOOP instruction.
b. Show the compiled code using delayed branch method.
c. Draw diagrams to show the timing of instruction pipelines for traditional pipeline, RISC pipeline with inserted NOOP, and RISC pipeline with delayed branch.

Answer:

(1) With inserted NOOP

Address	Instruction
100	LOAD R1←X
101	NOOP

102	ADD R1←R1+1
103	JUMP 107
104	NOOP
105	ADD R2←R2+R1
106	SUB R2←R2-R3
107	STORE Z←R1

(2)Delayed branch

Address	Instruction
100	LOAD R1←X
101	JUMP 105
102	ADD R1←R1+1
103	ADD R2←R2+R1
104	SUB R2←R2-R3
105	STORE Z←R1

(3)

Traditional pipeline

Address	Instruction	1	2	3	4	5	6	7	8
100	LOAD R1←X	I	E	D					
101	ADD R1←R1+1		I		E				
102	JUMP 105				I	E			
103	ADD R2←R2+R1					I			
105	STORE Z←R1						I	E	D

RISC pipeline with inserted NOOP

Address	Instruction	1	2	3	4	5	6	7	8
100	LOAD R1←X	I	E	D					
101	NOOP		I	E					
102	ADD R1←R1+1			I	E				
103	JUMP 107				I	E			
104	NOOP					I	E		
107	STORE Z←R1						I	E	D

RISC pipeline with delayed branch

Address	Instruction	1	2	3	4	5	6
100	LOAD R1←X	I	E	D			
101	JUMP 105		I	E			
102	ADD R1←R1+1			I	E		
105	STORE Z←R1				I	E	D