东南大学考试卷（ A 卷）

课程名称 计算结构（英文） 考试学期 17-18-2 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（15）	二（12）	三（10）	四（16）	五（47）	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、Fill blanks (Total 15 points, 1 points/blank)

1. Computer memory is organized into a hierarchy. List three types of internal memory in hierarchy from the top to down (Register) (Cache) (Main memory).
2. If (A)=5, (B)=3, (C)=16, and (D)=4, the result of **AB+CD//** is (2).
3. An instruction cycle can be divided into several stages, and, at least, each instruction cycle includes two basic stages: (Fetch) and (Execute).
4. The (Control Unit) is the engine that runs the entire computer. It controls everything with a few control signals to points within the processor and a few control signals to the system bus.
5. In (Direct\Indirect) addressing, the address field in the instruction refers to the address of a word in memory.
6. A computer has a cache and a main memory. 10 ns are required to load a word from the cache into the processor and, in the case of miss, additional 100 ns are required to load from the memory into the cache. If the hit ratio of the cache is 90%, the average word access time of the computer is (20) ns. If the hit ratio increases to 99%, the average word access time is (11) ns.
7. There are 200 registers in a computer, 8 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 (12), and the register window structure supports procedures with depth of (11) at most (with no need to save register variables to main memory).
8. (Machine parallelism) is a measure of the ability of the processor to take advantage of instruction-level parallelism.

9. One technique for implementing a control unit is referred to as (hardwired) implementation, in which the control unit is a combinatorial circuit, and an alternative control unit is a (microprogrammed) control unit.

二、 Select A, B, C, or D corresponding to the best answer from the given items (Total 12 points, 2 points/blank)

1. To perform a sequence of 16 instructions using pipeline, each instruction is divided into 5 stages, FI, DI, FO, EI and WO, where all the stages require the same cycle time. No branch is in the sequence. The speed up factor of the pipeline is (A).

   A. 4 B. 5 C. 16 D. 16/5

2. 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

3. An instruction has two address fields: source operand reference and destination operand reference, so the address of the next instruction is (D)

   A. The first address B. The second address
   C. Unknown D. Implicit

4. The instruction of 1 Byte length stored in X1 has an address mode of PC relative addressing and an address part X2 for operand reference. The effective address of the operand is (D)

   A. X1 B. X2 C. (X2) D. X1+X2+1

5. In the process of instruction pipeline operation with a branch predictor using dynamic 1-bit history information prediction, for a Loop of 1000 iterations, assuming that the initial bit for prediction is not-taken, the number of right predictions is (C).

   A. 1 B. 2 C. 998 D. 999

6. How many 32-bit integers can be stored in a cache line if the cache is 4KBytes, 4-way set-associative, and contains32 sets? (A)

   A. 8 B. 16 C. 32 D.64

三、Answer according to given contents (Total 10 points, 2 points/item)

Every processor or processor family has its own machine code instruction set. Instructions are patterns of bits that by physical design correspond to different commands to the machine. Thus, the instruction set is specific to a class of processors using (mostly) the same architecture. Successor or derivative processor designs often include all the instructions of a predecessor and may add additional instructions. Occasionally, a successor design will discontinue or alter the meaning of some instruction code (typically because it is needed for new purposes), affecting

code compatibility to some extent; even nearly completely compatible processors may show slightly different behavior for some instructions, but this is rarely a problem. Systems may also differ in other details, such as memory arrangement, operating systems, or peripheral devices. Because a program normally relies on such factors, different systems will typically not run the same machine code, even when the same type of processor is used.

A machine code instruction set may have all instructions of the same length, or it may have variable-length instructions. How the patterns are organized varies strongly with the particular architecture and often also with the type of instruction. Most instructions have one or more opcode fields which specifies the basic instruction type (such as arithmetic, logical, jump, etc.) and the actual operation (such as add or compare) and other fields that may give the type of the operand(s), the addressing mode(s), the addressing offset(s) or index, or the actual value itself (such constant operands contained in an instruction are called immediates).

Not all machines or individual instructions have explicit operands. An accumulator machine has a combined left operand and result in an implicit accumulator for most arithmetic instructions. Other architectures (such as 8086 and the x86-family) have accumulator versions of common instructions, with the accumulator regarded as one of the general registers by longer instructions. A stack machine has most or all of its operands on an implicit stack. Special purpose instructions also often lack explicit operands (CPUID in the x86 architecture writes values into four implicit destination registers, for instance). This distinction between explicit and implicit operands is important in code generators, especially in the register allocation and live range tracking parts. A good code optimizer can track implicit as well as explicit operands which may allow more frequent constant propagation, constant folding of registers (a register assigned the result of a constant expression freed up by replacing it by that constant) and other code enhancements.

a) Give a title for the above phases:

Answer: machine code instruction set\ machine instruction\ instruction set\ instruction

b) What is code compatibility?

Answer: The instruction set is specific to a class of processors using (mostly) the same architecture. Successor or derivative processor designs often include all the instructions of a predecessor.

c) “A machine code instruction set may have all instructions of the same length, or it may have variable-length instructions.” The variable-length of instructions results in dependency, what’s the dependency?

Answer: Procedure dependency

d) List the constitutions of instructions

Answer: One or more opcode fields, the type of the operand(s), the addressing mode(s), the addressing offset(s) or index, or the actual value itself.

e) How a stack machine accesses operands

Answer: Implicit

四、**Questions (Total 16 points)**

1. ①**(3 points)** What are characteristics of RISC? ②**(3 points)** The RISC architecture is a dramatic departure from the historical trend in processor architecture. What motivated the key characteristics of RISC machines?

   Answer:①

   - Large number of general purpose registers or use of compiler technology to optimize register use
   - Limited and simple instruction set
   - Emphasis on optimising the instruction pipeline

   Answer: ②

   Studies of the execution behavior of high-level language programs provide guidance in designing RISC:
   - Assignment predominate->simple movement of data should be optimized
   - Many IF and LOOP instructions->sequence control mechanism needs to be optimized for pipelining
   - Operand reference patterns->keeping a moderate number of operands in registers
2. ① **(2 points)** What are the basic tasks of a control unit of a processor? ② **(3 points)** How does a microprogrammed control unit perform these tasks?

   Answer:①

   - Sequence
   - Execute

   Answer: ②

   - Use sequences of instructions,
   - Microinstructions describes micro-operations; Control signals are generated by micro-operations

3. ① **(2 points)** What is **locality of reference**? ② **(3 points)** How to employ the principle of **locality of reference** in memory hierarchy?

Answer: ①

The memory references tend to cluster.

Answer: ②

It is possible to organize data across a memory hierarchy such that the percentage of accesses to each successively lower level is substantially less than that of the level above. Because memory references tend to cluster, the data in the higher-level memory need not change very often to satisfy memory access requests.

五、 Problems (Total 47 points)

1. **(9 points)** Given an address 0xA2EC4A8F (dressed in Byte), resolve the address to Word offset, Set/Line and Tag for the three caches in the following table, and fill the **hexadecimal** numbers corresponding to the resolution in the table:

Cache size (Bytes)	Block size (Bytes)	Mapping Policy	Tag	Set/ Line	Word offset
4096	4	Direct mapping	A2EC4	2A3	3
1024	16	4-way set associative	A2EC4A	8	F
2048	32	Associative mapping	5176254	None/ -/ Random	0F

2. **(4 points)** A stack is a part of the main memory, and its top element is pointed by the Stack Pointer (SP). Show the execute sequence of micro-operations for the instruction "**POP X**". This instruction removes the top item from the stack to memory employing direct addressing.

Answer:
t1: MAR←SP
t2: MBR←memory
SP←SP+1
t3: MAR←$IR_{address}$ （或 MAR←X）
t4: memory←MBR

3. A block diagram of the CPU with internal bus is shown in the following figure, in which, we can find two registers, labeled Y and Z.

a) **(2 points)** Please explain why the registers Y and Z are added to the organization.

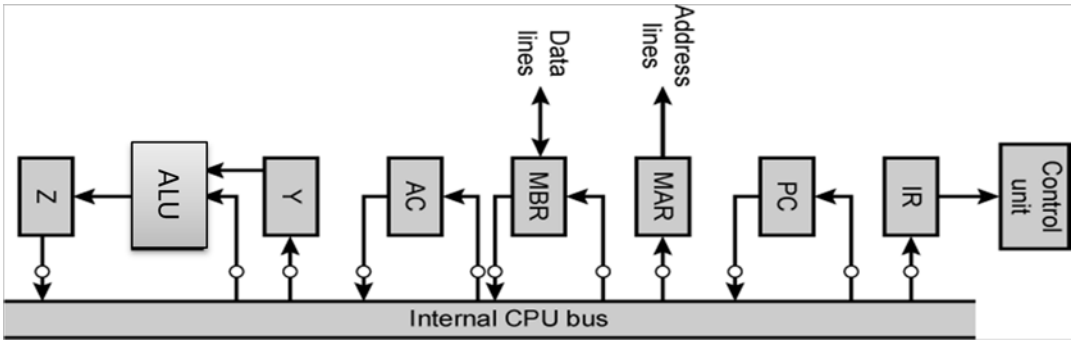
Answer:

The new registers are needed for the proper operation of the ALU.

Register Y provides temporary input storage, and Register Z provides temporary output storage.

b) **(6 points)** Write the sequence of micro-operations required for the bus structure of the following figure to add a number to the AC when the number is

- an immediate operand
- a direct-address operand
- an indirect-address operand



Answer:

a. an immediate operand: t1: Y←(IR(address))
t2: Z←(AC)+(Y)
t3: AC←(Z)

b. a direct-address operand: t1: MAR←(IR(address))
t2: MBR←Memory
t3: Y←(MBR)
t4: Z←(AC)+(Y)
t5: AC←(Z)

c. c. an indirect-address operand: t1: MAR←(IR(address))
t2: MBR←Memory
t3:MAR←(MBR)
t4: MBR←Memory
t5: Y←(MBR)
t6: Z←(AC)+(Y)
t7: AC←(Z)

4. Consider the following program

Address	Instruction
100	LOAD SR1←A; load A into symbolic register 1
101	LOAD SR2←B; load B into symbolic register 2
102	ADD SR3←SR1+SR2
103	LOAD SR4←C
104	ADD SR5←SR1+SR4
105	JUMP 107
106	SUB SR6←SR5-SR3
107	STORE Z← SR3

A compiler for a RISC machine can either insert NOOP instruction or do instruction reordering to deal with data dependencies among the instructions and branch instructions.

(1) **(3 points)** Show the compiled code using NOOP instruction.

Answer:

Address	Instruction
100	LOAD SR1←A
101	LOAD SR2←B
102	NOOP
103	ADD SR3←SR1+SR2
104	LOAD SR4←C
105	NOOP
106	ADD SR5←SR1+SR4
107	JUMP 110
108	NOOP
109	SUB SR6←SR5-SR3
110	STORE Z←SR3

(2) **(3 points)** Show the compiled code using instruction reordering method and delayed branch.

Answer:

Address	Instruction
100	LOAD SR1←A
101	LOAD SR2←B
102	LOAD SR4←C
103	ADD SR3←SR1+SR2
104	JUMP 107
105	ADD SR5←SR1+SR4
106	SUB SR6←SR5-SR3
107	STORE Z←SR3

(3) **(4 points)** Draw diagrams to show the timing of the instruction pipeline operation of the above (1) and (2), where each instruction may include stages of Fetch (F) and Execute (E), or include stages of Fetch (F), Execute (E) and Memory Read/Write (D). Assume that all the stages are with the same interval.

Answer: RISC pipeline with inserted NOOP

Address	Instruction	1	2	3	4	5	6	7	8	9
100	LOAD SR1←A	I	E	D						
101	LOAD SR2←B		I	E						
102	NOOP			I	E					
103	ADD SR3←SR1+SR2				I	E				
104	LOAD SR4←C					I	E	D		
105	NOOP						I	E		
106	ADD SR5←SR1+SR4							I	E	
107	JUMP 110								I	E
108	NOOP									I
110	STORE Z←SR3									

RISC pipeline with instruction reordering and delayed branch

Address	Instruction	1	2	3	4	5	6	7	8	9
100	LOAD SR1←A	I	E	D						
101	LOAD SR2←B		I	E	D					
102	LOAD SR4←C			I	E	D				
103	ADD SR3←SR1+SR2				I	E				
104	JUMP 107					I	E			
105	ADD SR5←SR1+SR4						I	E		
107	STORE Z←SR3							I	E	D

(4) **(2 points)** Calculate the speedup factor of the two pipelines of (3) compared with the instruction operation of no pipeline.

Answer:

Assume the time needed without pipeline is T1, the time needed for RISC pipeline with inserted NOOP is T2, the time needed for RISC pipeline with instruction reordering is T3, and the cycle time is c.

T1 = 4 * 3c + 3 * 2c = 18c.

T2 = 12c.

T3 = 9c.

The speedup factor for RISC pipeline with inserted NOOP is T1/T2 = 3/2.

The speedup factor for RISC pipeline with instruction reordering is T1/T3 = 2.

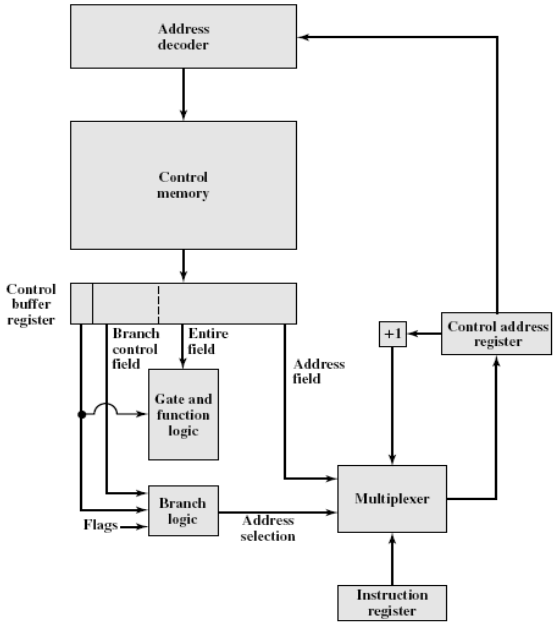
5. **(7 points)** Consider a 16-bit processor in which the following appears in main memory, starting at location 103:

103	Load to AC	Mode
104	750	
105	Next instruction	

The Mode field indicates an addressing mode, or a source register; assume that when used, the source register is R1, which has a value of 320. There is also a base register that contains the value 160. The value of 750 in location 104 may be part of the address calculation. Assume that location 301 contains value 901, location 302 contains value 902, and so on. The above numbers are decimal. Determine the effective address and the operand to be loaded for the following address modes and fill them to the blank:

a) Direct
b) Immediate
c) Indirect
d) PC relative
e) Displacement
f) Register
g) Register indirect

Addressing Mode	EA	Operand
a)	750	1350
b)	104	750
c)	1350	1950
d)	855	1455
e)	910	1510
f)	R1	320
g)	320	920



6. The above figure shows a microprogramed control unit using variable address microinstructions.

   a) **(2 points)** What's the format of variable address microinstrutions?

Answer:

- An indicating bit, one or more control fields, and one or zero address field /Address number is variable;
- A bit indicates that the microinstrution has or has not address field.

   b) **(2 points)** What's the advantage of variable address format?

Answer: Save bits for addressing in average

   c) **(3 points)** How to decide the address of the next microinstruction?

Answer:

The branch logic selects one address from the multiplexer and sends the address to control address register as the next microinstruction address. The address may be

- Get the next instruction (Add 1 to the control address register)
- Jump to a new routine based on a jump microinstruction, or
- Jump to a machine instruction routine