目　　錄

103 台大電機

1. 假設我們有一個序列執行程式，其百分之六十的執行時間，可以切割成四個相同的 thread 來執行。其餘百分之四十的執行時間，則完全無法平行化。請問，根據 Amdahl's law，這個程式經過轉換成四個 thread 的平行化程式，並在四核心的 CPU 上執行時，速度可以是原始序列執行程式的幾倍？
(a) 4 (b) 2.5 (c) 15/8 (d) 13/8 (e) 20/11

Answer: (e)

註：speedup = $\frac{1}{\frac{0.6}{4}+0.4}$ = 20/11 = 1.82

2. 假設我們有一個 CPU，執行一個 ALU 指令，要一個 cycle，一個 load 指令要三個 cycle，一個 branch 指令要五個 cycle。我們現在有一個程式 A，與甲、乙、丙三個 compiler，分別將 A 轉換成 A$_{甲}$、A$_{乙}$、A$_{丙}$三份機器碼(object code)。這三份機器碼執行時，分別會使用下列的指令數：

	ALU 指令數	Load 指令數	Branch 指令數
A$_{甲}$	80 萬	40 萬	50 萬
A$_{乙}$	90 萬	35 萬	48 萬
A$_{丙}$	40 萬	44 萬	58 萬

請問這三份機器碼執行的速度，由快而慢的順序是？
(a) A$_{甲}$A$_{乙}$A$_{丙}$ (b) A$_{丙}$A$_{乙}$A$_{甲}$ (c) A$_{甲}$A$_{丙}$A$_{乙}$ (d) A$_{丙}$A$_{甲}$A$_{乙}$ (e) A$_{乙}$A$_{甲}$A$_{丙}$

Answer: (e)

註： CPU clock cycles for A$_{甲}$ = 1 × 80 + 3 × 40 + 5 × 50 = 450 萬
CPU clock cycles for A$_{乙}$ = 1 × 90 + 3 × 35 + 5 × 48 = 435 萬
CPU clock cycles for A$_{丙}$ = 1 × 40 + 3 × 44 + 5 × 58 = 462 萬

3. 請問下列關於 Cache 記憶體的技術與理論，下列哪些敘述是正確的？ (複選)
(a) Cache 記憶體能發揮效能，是因為 locality of references。
(b) Cache 記憶體能發揮效能，是因為 Moore's law。
(c) Direct mapped cache 可以採用 best-fit 演算法，讓一個 block 在 run-time 放到最佳的 cache line 位置。
(d) 在 4-way set associate cache 中，每個 cache block 只能放在固定四個 memory block 中的一個。
(e) 在 miss ratio 是 0.1%時，在使用讀寫速度是記憶體十倍的 cache 後，讀寫速度可以是原來的九倍以上。

Answer: (a), (e)

註(d)：每個 memory block 只能放在固定四個 cache block 中的一個。

註(e)：假設 memory 的存取時間為 T。speedup = $\frac{T}{\frac{T}{10}+0.001T}$ = 9.9

4. 關於 cache management，以下敘述何者為真？

(a) read-through policy 在 read miss 時，沒有提供任何好處。

(b) write-through policy 在 read operation 時，有可能會產生記憶體的 write operation。

(c) write-back policy 在 cache block 被 replace 時，不需要將此 cache block 寫回記憶體。

(d) 在 write-miss 發生時，write-allocate policy 不會把要寫入的 cache block 放入記憶體。

(e) write-through policy 通常不會與 write-allocate policy 搭配使用。

Answer: (a), (e)

註(a)： Reads dominate processor cache accesses. The block can be read at the same time that the tag is read and compared, so the block read begins as soon as the block address is available. If the read is a miss, there is no benefit - but also no harm; just ignore the value read. The read policies are:

***Read Through* -** reading a word from main memory to CPU

***No Read Through* -** reading a block from main memory to cache and then from cache to CPU

5. 假設我們有一個電腦系統，使用 write-allocate cache。每個 cache block 有四個 words。CPU 每秒鐘會送出 10^7 個記憶體位址，其中 20%是 write operation。而 bus 也可以支援每秒 10^7 個 word 傳輸。假設在任何瞬間，30%的 cache blocks 都是 dirty。請問下列敘述，何者為真？

(a) 在採用 write-through policy 時，若 hit ratio (read 與 write 合計)是 0.95，則 bus 的 bandwidth 只使用了 25%。

(b) 在採用 write-through policy 時，若 hit ratio (read 與 write 合計)是 0.90，則 bus 的 bandwidth 只使用了 45%。

(c) 在採用 write-back policy 時，若 hit ratio (read 與 write 合計)是 0.95，則 bus 的 bandwidth 只使用了 26%。

(d) 在採用 write-back policy 時，若 hit ratio (read 與 write 合計)是 0.90，則 bus 的 bandwidth 只使用了 52%。

(e) 在採用 write-back policy 時，若 hit ratio (read 與 write 合計)是 0.88，則 bus 的 bandwidth 只使用了 72%。

Answer: none

註(a)：

	Bus Bandwidth Used	Reasoning
Read hit	0	Hit means reference is found in cache, so no bus bandwidth used
Read miss	$10^7 \times 0.05 \times 0.8 \times 4$	miss ratio = 1- hit ratio = 1 - 0.95 = 0.05 reads are 80% of total number of references block size = 4 words
Write	$10^7 \times 0.95 \times 0.2 \times 1$	Because we have write through policy we have to

hit		write to main memory on every hit. But we have to write only 1 word. Writes are 20% of total number of references, hit ratio = 0.95
Write miss	$10^7 \times 0.05 \times 0.2 \times 5$	On every write miss we have to write 1 word to and move 4 words from memory because write allocate policy. Writes are 20% of total number of references, hit ratio = 0.95

Total Bandwidth Used = BW used on Read hit + BW used on Read miss + BW used on Write hit + BW used on write miss = 0 + $10^7 \times 0.05 \times 0.8 \times 4 + 10^7 \times 0.95 \times 0.2 \times 1 + 10^7 \times 0.05 \times 0.2 \times 5 = 10^7 \times 0.4$. The percentage of the bus bandwidth used = 0.4

註(b)：Total Bandwidth Used = 0 + $10^7 \times 0.1 \times 0.8 \times 4 + 10^7 \times 0.9 \times 0.2 \times 1 + 10^7 \times 0.1 \times 0.2 \times 5 = 10^7 \times 0.6$. The percentage of the bus bandwidth used = 0.6

註(c)：

	Bus Bandwidth Used	Reasoning
Read hit	0	Hit means reference is found in cache, so no bus bandwidth used
Read miss	$10^7 \times 0.05 \times 0.8 \times (4 \times 0.3 + 4)$	miss ratio = 1- hit ratio = 1 - 0.95 = 0.05 reads are 80% of total number of references writes are 20% of total number of references block size = 4 words The term 4×0.3 refers to replacing the dirty block (0.3 is the probability of the block to be dirty). We write back the dirty block (4 words) and read needed block (another 4 words).
Write hit	0	Write hit does not generate any traffic on the bus, just makes the block in cache dirty.
Write miss	$10^7 \times 0.05 \times 0.2 \times 4$	On every write miss we have to move 4 words from memory because of write allocate policy. Writes are 20% of total number of references, hit ratio = 0.95

Total Bandwidth Used = BW used on Read hit + BW used on Read miss + BW used on Write hit + BW used on write miss = 0 + $10^7 \times 0.05 \times 0.8 \times (4 \times 0.3 + 4) + 0 + 10^7 \times 0.05 \times 0.2 \times 4 = 10^7 \times 0.248$. The percentage of the bus bandwidth used = 0.248

註(d)：Total Bandwidth Used = 0 + $10^7 \times 0.1 \times 0.8 \times (4 \times 0.3 + 4) + 0 + 10^7 \times 0.1 \times 0.2 \times 4 = 10^7 \times 0.496$. The percentage of the bus bandwidth used = 0.496

註(e)：Total Bandwidth Used = $0 + 10^7 \times 0.12 \times 0.8 \times (4 \times 0.3 + 4) + 0 + 10^7 \times 0.12 \times 0.2 \times 4 = 10^7 \times 0.595$. The percentage of the bus bandwidth used = 0.595

6. 為了發揮 pipeline 的效率，compiler 在產生機器碼(object code、machine code)時，可以把 for-loop 展開(unrolling)。請問下列敘述，何者為真？

(a) 展開後，可以避免在這個 loop 的機器碼尾端的 branch 指令造成的 pipeline stall。

(b) Compiler 會先把這個 loop 的機器碼產生出來，然後依據 loop 反覆次數，然後將該段機器碼，依序複寫數次，完成 unrolling 步驟。

(c) 這樣的展開，即可以獲得程式執行效率的提昇。

(d) 在 loop 中的資料變數(非 loop-maintenance 變數)存在相依性(dependency)，則 compiler 無法將此 loop 展開。

(e) 為了進一步提昇執行效率，compiler 也可能變動展開後的機器碼中指令執行的次序。

Answer: (a), (b), (c), (e)

註(a), (c)：Unrolled loop that minimizes stalls

7. 為了發揮程式機器碼在多核心(平行)計算機上的執行效率，compiler 可以將同一個 for-loop 的不同次的 loop-body 執行，配置到不同的核心上執行，達成 thread-level 的 parallelism。譬如下列程式：

```
for (i = 0; i < 100; i++) a[i] + 100;
```

- 對 $i \in [0, 24]$的 ”a[i] = a[i] + 100;”，可以放到第一個核心上序列執行;
- 對 $i \in [25, 49]$的 ”a[i] = a[i] + 100;”，可以放到第二個核心上序列執行;
- 對 $i \in [50, 74]$的 ”a[i] = a[i] + 100;”，可以放到第三個核心上序列執行;
- 對 $i \in [75, 99]$的 ”a[i] = a[i] + 100;”，可以放到第四三個核心上序列執行;

這樣，四個核心同時執行，就可以提昇執行速度到四倍。請問在此情況下，下列敘述何者為真？

(a) 下列 C 程式片段的機器碼可以未經 compiler 改寫，直接配置到四核心上平行執行。

```
for (i = 0; i < 100; i++) a[i] + *x;
```

(b) 下列 C 程式片段的機器碼可以未經 compiler 改寫，直接配置到四核心上平行執行。

```
for (i = 0; i < 100; i++) a[i] + a[0];
```

(c) 下列 C 程式片段的機器碼可以未經 compiler 改寫，直接配置到四核心上平行執行。

```
for (i = 0; i < 100; i++) a[i] + b[i];
```

(d) 下列 C 程式片段的機器碼可以未經 compiler 改寫，直接配置到四核心上平行執行。

```
for (i = 0; i < 100; i++) {a[i] = a[i] + a[i + 100]; a[i + 101] = a[i + 201];}
```

(e) 下列 C 程式片段的機器碼：

```
for (i = 0; i < 100; i++) {a[i] = a[i] + a[i + 100]; a[i + 101] = a[i + 201];}
```

可以經過 compiler 改寫成下列程式碼後，其 for-loop 可以直接配置到四核心上平行執行。

```
a[0] = a[0] + a[100]
```

```
for (i = 0; i < 99; i++) {a[i + 101] = a[i + 201]; a[i + 1] + a[i + 101];}
a[200] = a[300]
```

Answer: (a), (b), (c), (e)

8. 在 IEEE 754 的 single-precision floating point 表示法中，是用 32 個 bits 來表示一個浮點數。Most significant bit 是 sign bit，接下來八個 bits 是 biased 的 exponent，而最後的 23 個 bits 代表分數部份。請問在此表示法下，下列敘述何者為真？
(a) 00111111 11000000 00000000 00000000 代表 0.5
(b) 01000000 01010000 00000000 00000000 代表 3.25
(c) 10111111 01000000 00000000 00000000 代表-0.25
(d) 10111110 10010000 00000000 00000000 代表-0.28125
(e) 01000000 01101000 00000000 00000000 代表 1.625

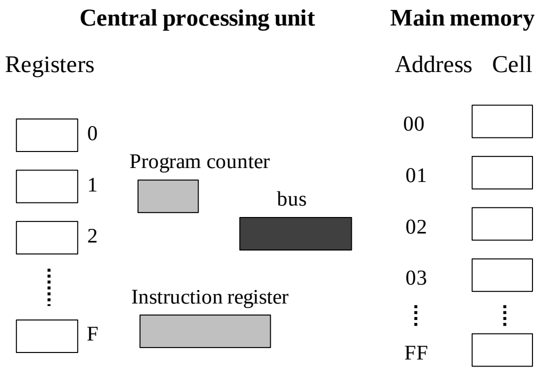
Answer: (b), (d)

註：

(a)	(b)	(c)	(d)	(e)
1.5	3.25	-0.75	-0.28125	3.625

9. 假設我們有下列的電腦，與其指令組(Instruction set)。

8-bit bus and 8-bit registers



12 machine instructions

Opcode	operand	Description
1	RXY	LOAD reg. R from cell XY.
2	RXY	LOAD reg. R with XY.
3	RXY	STORE reg. R at XY.
4	ORS	MOVE reg. R to S.
5	RST	ADD reg. S and T into R. (2's comp.)
6	RST	ADD reg. S and T into R. (floating pt.)
7	RST	OR reg. S and T into R.
8	RST	AND reg. S and T into R.
9	RST	XOR reg. S and T into R.
A	ROX	ROTATE reg. R X times.
B	RXY	JUMP to XY if reg. R = reg. 0.
C	000	HALT.

假設記憶體內有下列內容，而且 program counter 的內容是 0，請問這個電腦程式執行結束後，下列敘述何者為真？

記憶體位址(十六進位)	記憶體內容(十六進位)
00	24
01	08
02	13

03	12
04	20
05	00
06	21
07	FE
08	53
09	34
0A	54
0B	41
0C	B4
0D	10
0E	B0
0F	08
10	C0
11	00
12	1A

(a) reg. 0 的內容為 3 (十進位) (b) reg. 1 的內容為-2 (十進位) (c) reg. 2 的內容為 0
(d) reg. 3 的內容為 46 (十進位) (e) reg. 4 的內容為 0

Answer: (b), (d), (e)

註：

Mem. Address	Instruction	Register content (in dec.)
00	LOAD reg. 4 with 08.	R4 = 8
02	LOAD reg. 3 from cell 12.	R3 = 26
04	LOAD reg. 0 with 00.	R0 = 0
06	LOAD reg. 1 with FE.	R1 = -2
08	ADD reg. 3 and 4 into 3. (2's comp.)	R3 = 34, 40, 44, 46
0A	ADD reg. 4 and 1 into 4. (2's comp.)	R4 = 6, 4, 2, 0
0C	JUMP to 10 if reg. 4 = reg. 0.	$6 \neq 0$, $4 \neq 0$, $2 \neq 0$, (0 = 0 → jump to 10)
0E	JUMP to 08 if reg. 0 = reg. 0.	Jump to 08
10	HALT.	

10. 請問在區分 ISA(Instruction Set Architecture)的類別時，下列哪些敘述是正確的？
    (a) register-memory ISA 對 register 與記憶體(memory)的讀寫，採取嚴格分割。因此除了，load 與 store 指令外，其他指令不能直接讀寫記憶體中的資料變數。
    (b) 80x86 是 RISC 結構。
    (c) 所有的 MIPS 指令都是相同長度。
    (d) 80x86 因為是採取 RISC 結構，因此他的 branch instruction，如 BE、BNE，都只能檢驗 register 的內容，作為 branch 與否的依據。
    (e) MIPS 讀寫的物件位址須要是 aligned。

Answer: (a), (c), (e)

103 台聯大電機

1. Consider a virtual memory system with the following characteristics:
    - Page size of 64 KB = 2^{16} bytes
    - Page table and page directory entries of 8 bytes per entry
    - Inverted page table entries of 16 bytes per entry
    - 31 bit physical address, byte addressed
    - Two-level page table structure (containing a page directory and one layer of page tables)

    (1) What is the size (in number of address bits) of the virtual address space supported by the above virtual memory configuration?

    (2) What is the maximum required size (in KB, MB, or GB) of an inverted page table for the above virtual memory configuration?

Answer

(1) Page size of 64 KB gives 2^{16} address space = 16 bits

Each Page Table has 64 KB/8 bytes of entries = $2^{16}/2^{3}$ = 13 bits of address space

The Page Directory is the same size as a page table, and so contributes another 13 bits of address space.

16 + 13 + 13 = 42 bit virtual address space.

(2) There must be one inverted page table entry for each page of physical memory.

$2^{31}/2^{16} = 2^{15}$ pages of physical memory, meaning the inverted page table must be $2^{15} \times 16$ bytes = $2^{15} \times 2^{4}$ bytes = 0.5 MB in size

2. Consider two different implementations. M1 and M2, of the same instruction set. There are four classes of instructions (A, B, C and D) in the instruction set. M1 has a clock rate of 100 MHz and M2 has a clock rate of 150 MHz. The average number of cycles for each instruction class and their frequencies (for a typical program) are as follows:

Instruction Class	Machine M1-Cycles/ Instruction Class	Machine M2-Cycles/ Instruction Class	Frequency
A	1	2	40%
B	2	3	10%
C	4	4	30%
D	6	6	20%

(1) Calculate the average CPI for each machine. M1 and M2.

(2) Calculate the average MIPS ratings for each machine, M1 and M2.

(3) Which machine has a smaller MIPS rating? Which individual instruction class CPI do you need to change, and by how much, to have this machine have the same or better performance as the machine with the higher MIPS?

Answer

(1) $\text{CPI}_{\text{M1}} = 1 \times 0.4 + 2 \times 0.1 + 4 \times 0.3 + 6 \times 0.2 = 3$

$CPI_{M2} = 2 \times 0.4 + 3 \times 0.1 + 4 \times 0.3 + 6 \times 0.2 = 3.5$

(2) $MIPS_{M1} = (100 \times 10^6) / (3 \times 10^6) = 33.33$

$MIPS_{M2} = (150 \times 10^6) / (3.5 \times 10^6) = 42.86$

(3) Machine M1 has a smaller MIPS rating. If we change the CPI of instruction class D for Machine M1 to 2, we will have a better MIPS rating than M1:

$CPI_{M1} = 1 \times 0.4 + 2 \times 0.1 + 4 \times 0.3 + 2 \times 0.2 = 2.2$

Average MIPS rating = $(100 \times 10^6) / (2.2 \times 10^6) = 45.45$

3. Assume we are designing a 16-bit MIPS CPU with 16-bit instruction words.

(1) Assume the IEEE-754 floating-point representation is also adjusted to 16-bit long for this 16-bit MIPS CPU. If the exponent is 6-bit and the mantissa is 9-bit in this 16-bit floating-point format, what is the maximum number that it can represent? Please use base-10 scientific notation to show your answer, (assume $\log_{10}2 = 0.3$)

(2) In the IEEE-754 floating-point representation, the exponent field employs the excess-N coding. What should be *N* in your 16-bit floating-point format if it follows the similar mechanism?

(3) Given two decimal numbers 6.25 and -3.375. Please translate them into the 16-bit floating-point format specified in (1) and calculate the sum of the two numbers using floating-point addition. Assume that we can store only four digits of the mantissa fields during the addition. Because no extra guard and round bits are available, the digits exceeding 4 bits will be truncated.

Answer

(1) $+1.111111111 \times 2^{31} \approx +2 \times 10^{31\times0.3} = +2 \times 10^{9.3}$

(2) N = 31

(3) $6.25 = 110.01 = 1.1001 \times 2^2$, $-3.375 = -11.011 = -1.1011 \times 2^1$

step1: $-1.1011 \times 2^1 = -0.1101 \times 2^2$

step2: $(1.1001 - 0.1101) \times 2^2 = 0.1100 \times 2^2$

step3: $0.1100 \times 2^2 = 1.1000 \times 2^1$ → neither overflow nor underflow

$1.1000 \times 2^1 = 11 = 3_{10}$

4. Assume variable sum is passed with register $a0. Therefore, the following C program segment can be converted to the MIPS assembly code as shown below.

```
int sum (int n) {
      if (n < 1) return 0;
     else return (n + sum(n-1));
 }
```

```
Sum:   addi  $sp, $sp, (a)
       sw    $ra, 4($3p)
       sw    $a0, 0($sp)
       slti  $t0, $a0, 1
       beq   $t0, $zero, L1
       add   $v0, $zero, (b)
       addi  $sp, $sp, 8
       jr    $ra
L1:    addi  $a0, $a0, -l
       jal   Sum
       lw    $a0, 0($sp)
       lw    $ra, 4($sp)
       addi  $sp, $sp, 8
       (c)   $v0, $a0, $v0
       jr    $ra
```

(1) Please fill in the blanks (a), (b), (c) to complete this assembly code.

(2) Assume the initial values of $a0, $t0, $v0 are 5, 6, 7, respectively. What is the final value of $v0 after this code is finished?

(3) Please translate the instruction lw $ra, 4($sp) to machine code in hexadecimal form. (lw → opcode=35, $sp → 29, $ra → 31)

(4) Assume the current PC value is 0x20406080 while starting the execution of this code, what values should be filled into the address fields of the instructions beq $t0, $zero, L1 and jal Sum respectively?

Answer

(1)

(a)	(b)	(c)
-8	$zero	add

(2) $v0 = 15

(3) 100011 11101 11111 0000000000000100 → 8FBF0004$_{hex}$

(4)

address field of beq	address field of jal
0000000000000011	00000100000001100000100000

5. Please describe the following terms concerning computer architecture

(1) What are the instruction-level parallelism (ILP), out-of-order (OOO) execution, and multi-issue processor? How to guarantee precise interrupt in a multi-issue superscalar processor?

(2) Please describe at least four different ways to update the PC in the simplified MIPS. Both an exception and a jal instruction can change the program flow and then return. What is the major difference between them?

(3) Why should the operands of arithmetic instructions come from registers instead of memory in modern processors? Why should the number of available registers be limited? Briefly describe at least 3 reasons.

Answer

(1) **Instruction-level parallelism:** the parallelism among instructions.

Out-of-order execution: a situation in pipelined execution when an instruction blocked from executing does not cause the following instructions to wait.

Multi-issue processor: a processor which multiple instructions are launched in one clock cycle.

If the saved process state is consistent with the sequential architectural model, then we can guarantee precise interrupt in a multi-issue superscalar processor.

(2)
1. Exception - e.g., ALU overflow
2. Interrupt - e.g., I/O request
3. Fault - e.g., page fault
4. Trap - e.g., operating system call
5. Flow control instruction execution - e.g., branches and jumps

An exception change the PC register to point to the beginning of the exception handler routine (part of OS code) while saving the old PC so that normal program flow can be restored later.

A JAL instruction also saving the old PC and change the PC register but to point to the beginning of a called procedure rather than an OS code.

(3) Memory is slower than registers, since there are fewer registers. Data (operands) accesses are faster if data is in registers instead of memory.

1. Memory is slower than registers, since there are fewer registers. Hence, data (operands) accesses are faster if data is in registers instead of memory.
2. A MIPS arithmetic instruction can read two registers, operate on them, and write the result. A data transfer instruction only reads one operand or writes one operand, without operating on it. Thus, registers take less time to access and have higher throughput than memory.
3. Accessing registers also uses less energy than accessing memory.

6. With pipelining, many instructions are simultaneous executing in a single datapath in every clock cycle. Consider the following instruction sequence:

```
lw    $t1, 0($t0)
lw    $t2, 4($t0)
add   $t3, $t1, $t2
sw    $t3, 12($t0)
lw    $t4, 8($t0)
add   $t5, $t1, $t4
sw    $t5, 16($t0)
```

For the classical MIPS with 5-stage pipeline datapath,

(1) Find the hazards in the given code segment.

(2) How many cycles are required to complete the code segment, if you have no forwarding path? And, how many cycles are required to complete the code, if you have implemented necessary forwarding paths.

(3) Try to reorder the code sequence to avoid stalls. Please write down the new code sequence.

Answer

(1)	(2)	(3)

(I_1, I_3) for t1;
(I_2, I_3) for t2;
(I_3, I_4) for t3;
(I_5, I_6) for t4;
(I_6, I_7) for t5.
	Without forwarding:
	(5 – 1) + 7 + 8 = 19
	With forwarding:
	(5 – 1) + 7 + 2 = 13
		lw $t1, 0($t0)
		lw $t2, 4($t0)
		lw $t4, 8($t0)
		add $t3, $t1, $t2
		add $t5, $t1, $t4
		sw $t3, 12($t0)
		sw $t5, 16($t0)

7. Please design a GCD (Greatest Common Divisor) calculator to calculate two numbers kept in registers (says R0 and R1), and place the result in the third register (says R2).

(1) Define the types and formats of instructions required for your GCD calculator. Explain the instruction formats you defined.

(2) Implement the GCD calculator with the assembly instructions you defined in (1).

(3) Implement all the instructions you defined in (1) using the basic building blocks (register file, memory, ALU, adder, MUX, etc).

(4) Continuing on (3), illustrate the datapath of your GCD calculator.

Answer

(1) Instruction format:

R-format:	OP	Rs1	Rs2	Rd
I-format:	OP	Rs1	Rs2	address

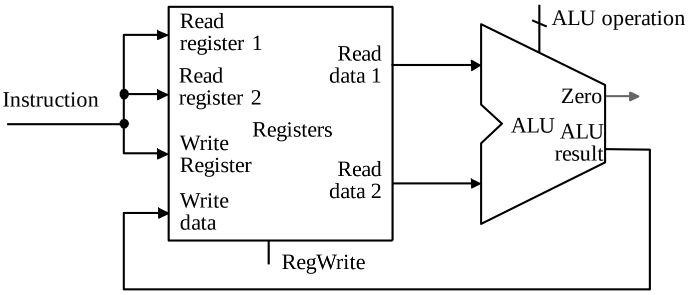
Instruction types:

Instruction	Function	Format
add Rd, Rs1, Rs2	Rd 🡐 Rs1 + Rs2	R-format
sub Rd, Rs1, Rs2	Rd 🡐 Rs1 - Rs2	R-format
slt Rd, Rs1, Rs2	If (Rs1 < Rs2) Rd 🡐 1 else Rd 🡐 0	R-format
beq Rs1, Rs2, Loop1	If (Rs1 = Rs2) goto Loop1	I-format

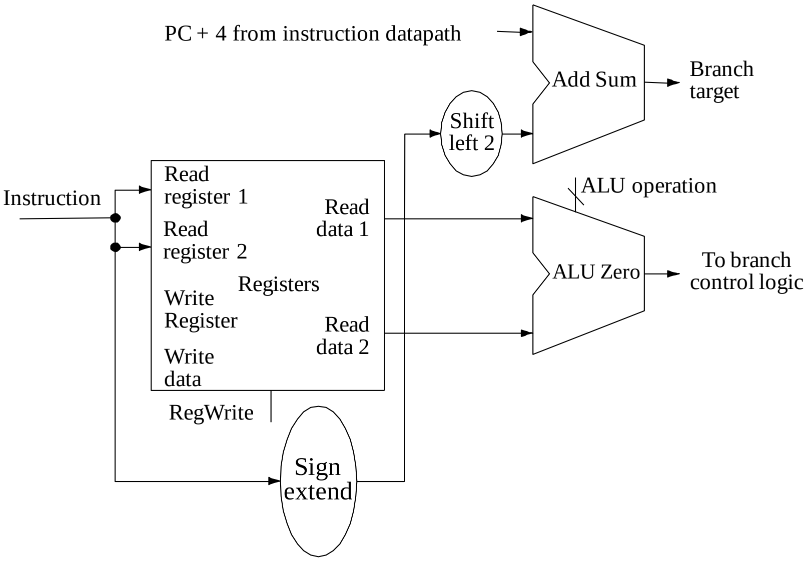
(2) Suppose the content of register Rz is equal to 0

```
Loop1:  beq   R0, R1, exit
        slt   R3, R0, R1
        beq   R3, Rz, else
        sub   R1, R1, R0
        beq   Rz, Rz, Loop1
else:   sub   R0, R0, R1
        beq   Rz, Rz, Loop1
exit:   add   R2, R0, Rz
```

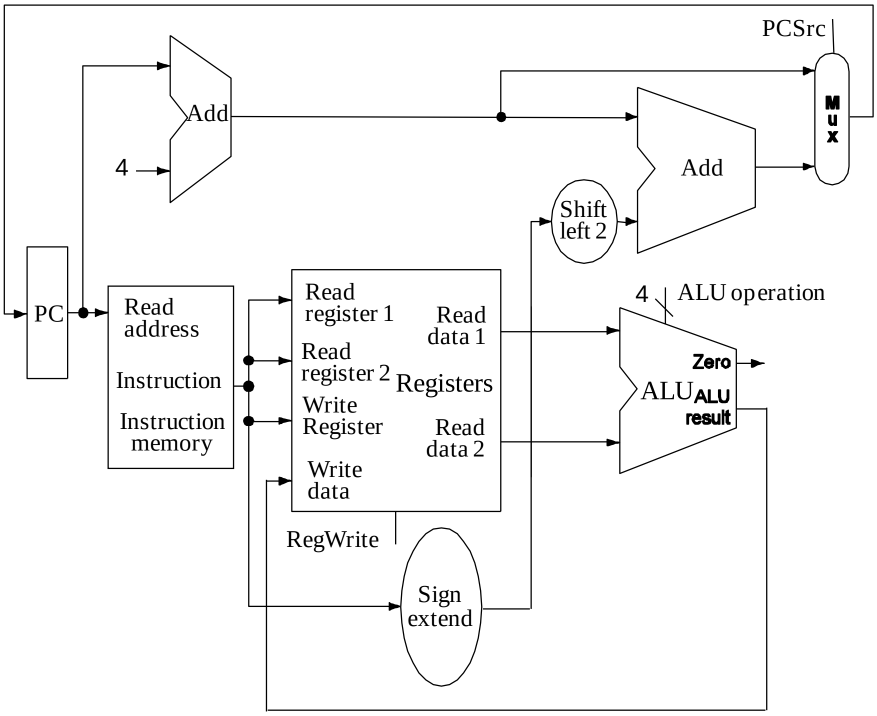
(3) R-format instruction datapath



Beq instruction dtapath



(4) GCD calculator datapath



註(2):兩整數的最大公因數可用輾轉相除法求出，演算法如下：

```
while (X != Y) {
    if (X < Y)
        Y = Y – X;
    else
        X = X – Y;
}
GCD = X;
```

103 清大資工

1. A computer has 3 instruction classes. They are A, B and C. The A instruction class is 1 CP1 (clock cycles per instruction), the B instruction is 2 CPI and the C instruction is 3 CPI. A program code has 5 millions of the A instruction class, 2 millions of the B instruction class and 3 millions of the C instruction class. Assume that the clock rate of the computer is 100 MHz. What is the execution time of the program code?

Answer

Execution time = $\frac{(5\times1+2\times2+3\times3)\times10^6}{100\times10^6}$ = 0.18 sec.

2. What is the decimal value of the 4-bit two's complement binary number 1010?

Answer

$1010_2 = -2^3 + 2^1 = -6_{10}$

3. (a) A full adder has two 1-bit numbers x_i, y_i, a carry-in bit c_i, a sum bit z_i and a carry-out bit c_{i+1}. Write the equations of the sum bit z_i in terms of x_i, y_i, and c_i.

(b) A 4-bit carry-look-ahead adder has two 4-bit binary numbers $x_3x_2x_1x_0$ and $y_3y_2y_1y_0$, a carry-in bit c_0, the sum bits $z_3z_2z_1z_0$ and a carry-out bit c_4. It can be formed from 4 stages, each of which is a full adder by replacing its output carry line c_i, by two carry generate signals g_i and p_i, where $0 \le i \le 3$. Write the equation for the carry-out bit c_4 of the 4-bit carry-look-ahead adder in terms c_0, g_i and p_i, where $0 \le i \le 3$.

Answer

(a) $z_i = x_i\overline{y_i}\overline{c_i} + \overline{x_i}y_i\overline{c_i} + \overline{x_i}\overline{y_i}c_i + x_iy_ic_i$

(b) $c_4 = g_3 + p_3g_2 + p_3p_2g_1 + p_3p_2p_1g_0 + p_3p_2p_1p_0c_0$

4. In a data cache, consider what the cache must do on a write miss.

(a) Describe two options for a write-through cache in response to a write miss.

(b) Describe a scenario that can benefit from each option listed in Part (a) above and briefly explain why.

(c) Describe the typical option for write-back cache in response to a write miss. Why are there fewer options than write-through caches?

Answer

(a) The first option is to allocate a block in the cache, called **write allocate**. The block is fetched from memory and then the appropriate portion of the block is overwritten.
The second option is to update the portion of the block in memory but not put it in the cache, called **no write allocate**.

(b) The motivation for **no write allocate** is that sometimes programs write entire blocks of data, such as when the operating system zeros a page of memory. In such cases, the fetch associated with the initial write miss may be unnecessary.

(c) The typical option for write-back cache in response to a write miss is **write allocate**. Since in write-back cache, no block copy resides in memory, if only update the portion of the block in memory that will results in the updated data in cache block is inconsistent with the same block in memory.

5. If there is no mechanism for cache invalidation or for marking certain memory locations non-cacheable, the data cache and the DMA controller may not work properly together.
   (a) If the cache is a write-back cache, can something go wrong if the DMA controller performs a read? What about a write?
   (b) If the cache is a write-through cache, can something go wrong if the DMA controller performs a read? How about a write?

Answer

(a) Consider a read from I/O that the DMA unit places directly into memory in a write-back cache. If some of the locations into which the DMA writes are in the cache, the processor will receive the old value when it does a read.

On the other hand, the DMA may write a value from memory to I/O when a newer value is in the cache, and the value has not been written back. That will result in the I/O gets the old value.

(b) In write-through cache, a read from I/O may also result in the processor receives the old value but a write to I/O doesn't cause the I/O receives the old value since all the newer data in cache will update immediately to the memory block.

6. Given a basic five-stage pipelined MIPS CPU with comparator with the branch address computing done at the ID stage and the following sequence of instructions. We assume that branch predictor always predicts next instruction and for the execution of *#40* beq the prediction is not taken.

```
36: lw   $1, 8($4)       72: lw   $4, 50($7)
40: beq  $1, $3, 7       76: add  $3, $4, $3
44: and  $12, $2, $5     80: or   $3, $2, $6
     ….                  84: slt  $3, $6, $3
```

Please answer the following questions for execution from instruction #36 to #84.

(a) Assume there is no forwarding in this pipelined processor. How many NOPs are inserted?
(b) Assume full forwarding. How many NOPs are inserted?

Answer

(a) 6 NOPS. 2 between (36, 40), 2 between (72, 76), and 2 between (80, 84)

(b) 3 NOPS. 2 between (36, 40), 1 between (72, 76)

7. True or False

(a) Like SMP, message-passing computers rely on locks for synchronization.

(b) Both multithreading and multicore rely on parallelism to get more efficiency from a chip.

(c) GPUs rely on graphics DRAM chips to reduce memory latency and thereby increase performance on graphics applications.

(d) Shared memory multiprocessors cannot take advantage of job-level parallelism.

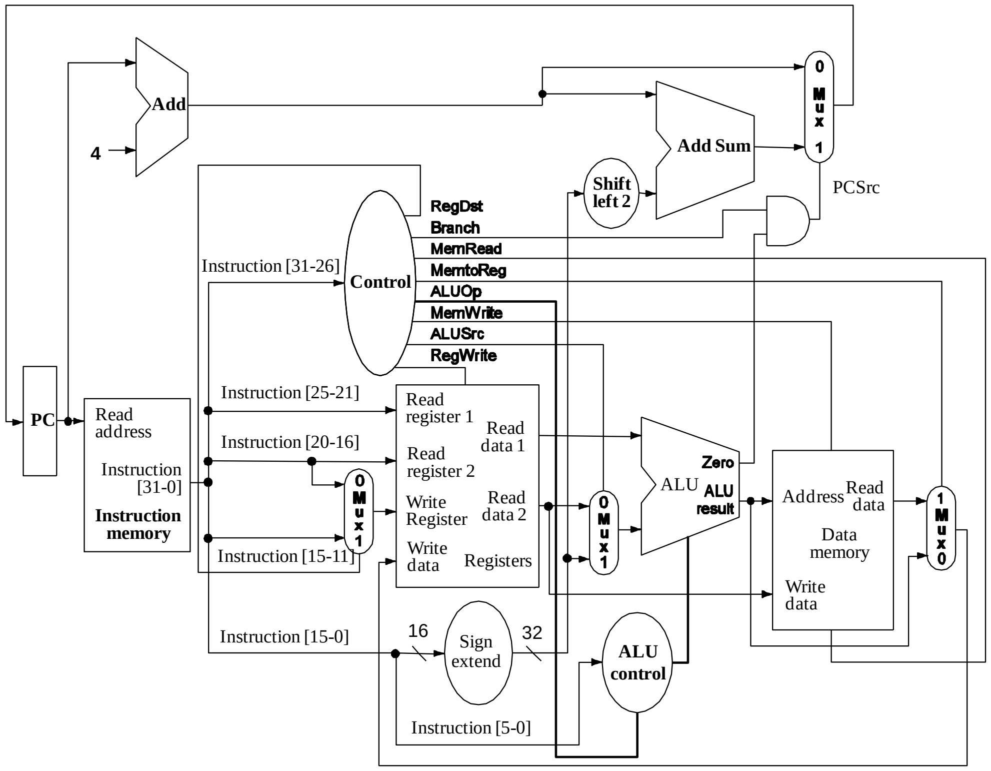
Answer

(a)	(b)	(c)	(d)
F	T	F	F

註(c)：GPUs rely on having enough threads to hide the latency to memory.

8. What are the values (0 or 1) of control signals generated by the control in the following graph?

Instruction	ALUSrc	RegWrite	MemRead	MemWrite	Branch
beq Rs, Rt, L					



Answer

Instruction	ALUSrc	RegWrite	MemRead	MemWrite	Branch
beq Rs, Rt, L	0	0	0	0	1

103 交大資聯

單選題

1. A 32bit multiplication can be designed as a sequential version (A) in the below figure. The same multiplication function can be achieved by a reduced hardware version (B). In particular, since when a shift is usually faster than addition, a Booth's algorithm is to find a string of 1's in multiplier and to replace he sequential additions with an initial subtract from Multiplicand when we see a first 1 and then later add Multiplicand for the test bit after the last 1. Which one of the following statements is *incorrect*?

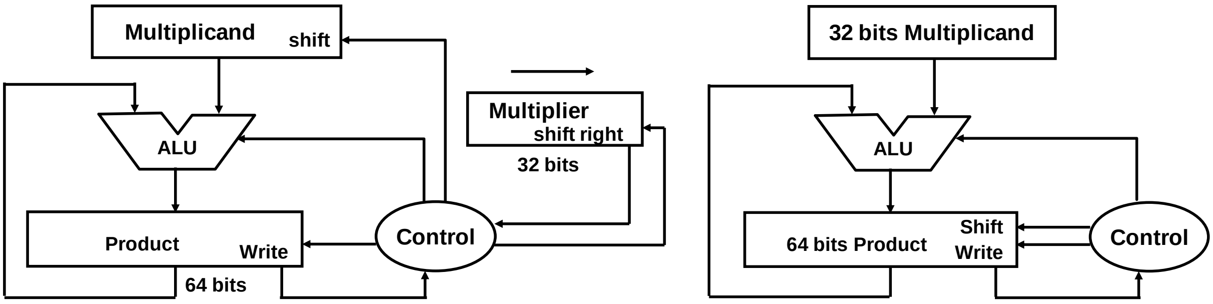
   (a) The Multiplicand of version (A) has to be a 64-bit register.

   (b) The product register of version (B) has to perform a logical shift right.

   (c) The product register of version (B) has to be initialized with the Multiplier.

   (d) The ALU of version (B) has to be a 64-bit adder (can also perform subtract).

   (e) For a multiplier value 0X00FF0F00, a Booth multiplication will perform only 2 adds, 2 subtracts, and 32 shifts



(A) Sequential version of multiplication (B) Reduced version with Booth's multiplication

Answer: (d)

2. A load-use hazard is that the data being loaded by a load instruction is not available yet when it is needed by another dependent instruction. When the hazard detection unit in 5 pipeline stages (IF, ID, EX, MEM, and WB) of MIPS CPU detects a load-use hazard, the CPU will stall for one cycle. Which one of the following statements is *incorrect*?

   (a) The detection condition is
   IF (ID/EX.MemRead && ((ID/EX.RegisterRt = IF/ID.RegisterRs) || (ID/EX.RegisterRt = IF/ID.RegisterRt)))

   (b) Once the hazard is detected and the CPU is stalled, the forwarding logic will be no longer needed.

   (c) Once the hazard is detected, the PC register will be disabled for a new write.

   (d) Once the hazard is detected, the IF/ID register will be disabled from changes.

   (e) The pipeline stages starting from EX stage must be flushed by setting all control signals to 0 (disabling controls).

Answer: (b)

3. Which one of the following statements is correct?
   (a) The main reason why CPU’s working frequency does not increases significantly these years is that CPU designers focus to push CPU architecture to multi-core platform.
   (b) The consumed power of a circuit increases linearly with the used working voltage.
   (c) The CPU operating clock rate has not been continuously and significantly improved these years because Moor’s law has become invalidate.
   (d) Consider the issues of integrated circuit manufacturing, manufacturing yield increases as die area decreases.
   (e) The cost per die decreases as manufacturing yield decreases.

Answer: (d)

註(a)：The major reasons include power wall and gap between processor and memory.

註(b)：The consumed power of a circuit increases linearly with voltage^2.

4. Which one of the following statements is *not* correct?
   (a) Induction-variable access is an example of temporal locality.
   (b) Array access is an example of spatial locality.
   (c) For a fixed-size cache, initially larger block can reduce miss rate due to spatial locality; however, as block size increases to a certain level, miss rate may increase.
   (d) For cache access scheme, write-through scheme has to write data to memory and data consistency is realize, but CPU has to wait.
   (e) Write buffer scheme is similar to write-through except that write buffer scheme writes data to buffers rather than to memory. CPU also needs to wait for the completion of buffer writing but does not need to wait too long since buffer writing is much faster than memory writing.

Answer: none

註： An induction variable is a variable that gets increased or decreased by a fixed amount on every iteration of a loop. For example, in the following loop, i and j are induction variables:

```
for (i=0; i < 10; ++i) {
    j = 17 * i;
}
```

複選題

5. You are so smart to find out an important observation that most branch instructions simply compare one register with zero or non-zero. Now we no longer provide beq rs, rt, target and instead you change the MIPS processor design so that we only allow one register operand for branch compare instruction:

   Bez r3, 1000 means if r3 == 0, then PC = (PC + 4) + 4000

   Because comparing with zero is easier than a subtract comparison, now the outcome of a branch can be determined earlier in the EX stage. Which of the following statements are correct for your change?

   (a) The new design can contribute better performance than the original MIPS design (determined in the MEM stage).

   (b) If we take “assume-branch-not-taken” prediction to handle the branch stall and bez branches are really untaken half the time, the recover mechanism would be simply to flush instructions in the IF, ID, and EX stages of the pipeline.

   (c) The 2-bit branch prediction will have more accurate prediction rate.

   (d) If the bez instruction is implemented as a delayed branch, there will be 2 delayed slots for compiler to schedule instructions.

   (e) It would take less hardware cost to move the bez branch resolution earlier to the ID stage, as we don’t need to provide forwarding logic for the bez.

Answer: (d)

註(a)： The new design may have less control hazard penalties but require more instructions and may not have better performance than the old one.

註(b)： Only the instructions at IF and ID stage should be flushed.

註(e)： Forwarding logic is still needed.

6. Following the same design in the above question, now your boss requires you to evaluate the real performance for the new design. We know that the cache has the longest latency in the pipeline and the original MIPS design determines branches in the MEM stage. Assume that originally 40% of the total instructions are beq, where 75% of them compare one register with zero. Other 25% original “beq” has to be changed:

   beq r3, r5, 1000 is changed into sub r1, r3, r5
   bez r1, 1000

   Consider the law of performance as $\frac{\text{instrucitons}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{time}}{\text{cycle}}$, which of the following statements are *correct* for your change?

   (a) The clock rate of the new design will be better, as comparing with zero is simple.

   (b) The new number of the instructions will be 1.1 times more than the one in the original design.

   (c) Assume no dynamic prediction and no assume-branch-not-taken prediction are provided, the CPI of your new design will be 1.9.

(d) Same as the above (without any prediction), the overall speedup over the original design will be no more than 1.16.

(e) The compiler has no extra overhead to schedule the new instructions, as the branch function is equivalent.

Answer: (b), (c), (d)

註(b)： 1 + 0.4 × 0.25 = 1.1

註(c)： CPI = 1 + 0.4 × 0.25 + 0.4 × 2 = 1.9

註(d)： Speedup = (1 + 0.4 × 3) / 1.9 = 1.158

7. Which of the following statements are *not* correct?

(a) If a disk manufacturer quotes MTTF as 90 years, it means the manufacturer claims that each disk produced by the manufacturer can operate normally for 90 years in average.

(b) RAID 0 duplicates data to multiple disks, which improves access performance.

(c) If the disk to be accessed fails, RAID 1 will reads data from a mirroring disk.

(d) For RAID 3, a data is split at block level.

(e) RAID 4 needs to read all disks for a read access since data has been split and distributed to redundant disks.

Answer: (b), (d), (e)

題組

A. Consider 32-bit byte address, a direct-mapped cache of size 64K bytes with each block of size 4 words. A series of memory accesses occur as follows (assume all states are reset initially): 0XDF105670, 0XDF10567C, 0X23A033A0, 0X23A034A0, 0XDF105678.

8. The number if tag bits is (a) 12 bits, (b) 16 bits, (c) 20 bits, (d) 8 bits.

Answer: (b)

註： Number of blocks = 64KB / 16 B = 4K → length of index field = 12 bits

The number of tag bits = 32 – 12 – 4 = 16

9. The number of blocks in the cache is (a) 4K, (b) 2K, (c) 1K, (d) 8K.

Answer: (a)

10. The hit access is

(a) Only 0XDF10567C.

(b) Only 0X23A034A0.

(c) 0XDF10567C and 0XDF105678.

(c) 0XDF10567C, 0X23A034A0, and 0XDF105678.

Answer: (c)

Address	Tag	Index	Hit/Miss
DF105670	DF10	567	Miss
DF10567C	DF10	567	Hit
23A033A0	23A0	33A	Miss
23A034A0	23A0	34A	Miss
DF105678	DF10	567	Hit

B. Two different compliers A and B are employed to compile a program that will be run on a machine with four types of instructions. The CPI of four types of instructions and the instruction counts (IC) produced by two compilers are listed below. The code sequence 1 requires P_0 clock cycles for execution while the code sequence 2 requires P_1 clock cycles for execution. If we want to reduce by 50% the required clock cycles for executing code sequence 1 by lowering CPI of class C instruction and remaining the CPIs of the other instructions unchanged, the CPI of class-C instruction should be lowered to *cp*.

Class	A	B	C	D
CPI for class	2	3	5	1
IC in sequence 1 by Compiler A	4	4	9	7
IC in sequence 2 by Compiler B	8	4	6	8

11. P_0 = (a) 71, (b) 66, (c) 72, (d) 70, (e) 60.

Answer: (c)

註：$P_0 = 2 \times 4 + 3 \times 4 + 5 \times 9 + 1 \times 7 = 72$

12. P_1 = (a) 71, (b) 66, (c) 72, (d) 70, (e) 60.

Answer: (b)

註：$P_1 = 2 \times 8 + 3 \times 4 + 5 \times 6 + 1 \times 8 = 66$

13. *cp* = (a) 2, (b) 1.5, (c) 3, (d) 1, (e) no solution.

Answer: (d)

註：$2 \times 4 + 3 \times 4 + cp \times 9 + 1 \times 7 = 72 \times 0.5 \rightarrow cp = 1$

103 成大電機

Select the most appropriate answers for the following multiple choice questions (1 to 7). Each question may have more than one answer. 10 point each, no partial point, no penalty.

1. Single-Instruction Multiple-Thread (SIMT) microarchitectures implemented in Graphics Processing Units (GPUs) run fine-grained threads in lockstep by grouping them into units, referred to as warps, to amortize the cost of instruction fetch, decode and control logic over multiple execution units. As individual threads take divergent execution paths, their processing takes place sequentially, defeating part of the efficiency advantage of SIMD execution. Which of the following statements is (are) true?
   (a) An SIMT thread runs on the microarchitecture in GPU.
   (b) A group of SIMT threads which are executed in lockstep are referred to as a warp.
   (c) "As individual threads take divergent execution paths," this refers to the execution of a conditional instruction.
   (d) "Defeating part of the efficiency advantage of SIMD execution," this means SIMT has better performance than SIMD.

Answer: (b)

註(a)：A group of SIMT threads (called warp) runs on the microarchitecture in GPU.

註(d)： On overall performance, SIMD is still greater than SIMT.

2. For CPUs, the problem of exception support was solved at a relatively early stage. This support was a key enabler to their success, and instrumental in this success was the definition of precise exception handling, where an exception is handled precisely if, with respect to the excepting instruction, the exception is handled and the process resumed at a point consistent with the sequential architectural model. Which of the following statements is (are) true?
   (a) "the exception is handled and the process resumed at a point consistent with the sequential architectural model," this means handling the exception by executing the instruction behind the excepting instruction.
   (b) "the exception is handled and the process resumed at a point consistent with the sequential architectural model," this means resuming instruction execution at the point behind the excepting instruction.
   (c) When multiple exceptions occur at the same time, there is only one PC which is stored as the exception PC.
   (d) The excepting instruction above is also referred to as the faulting instruction.

Answer: (b), (c), (d)

註(d)：Faulting instruction: instruction that caused an exception.

3. Which of the following is (are) true for a cache of 128 KB and 64-byte line size?
   (a) If the cacheable address space is 8GB, the tag width uses 10 bits for a direct-mapped structure.
   (b) If the cacheable address space is 4GB, the index width uses 10 bits for a 2-way set associative structure.
   (c) If the cacheable address space is 2GB, the line size width bits uses 6 bits for a direct-mapped structure.
   (d) If the cacheable address space is 16GB, the tag width uses 19 bits for a 4-way set associative structure.

Answer: (b), (c), (d)

註(a)：tag width = 33–11–6 = 16

註(d)：tag width = 34–9–6 = 19

4. Which of the following is (are) true?
   (a) The target instruction address of an indirect jump is not known until run time.
   (b) The target instruction address of an indirect jump is known at the compile time.
   (c) The target of instruction of jal SUB is computed by the programmer.
   (d) jr ra is an indirect jump operation.

Answer: (a), (d)

註(d)：jr belongs to register-Indirect Jump

5. Which of the following is (are) true?
   (a) A computer fetches its first instruction from its hard disk.
   (b) Cache is pronounced as /keɪʃ/.
   (c) Cache is pronounced as /kæ tʃ/.
   (d) Cache is pronounced as /keɪtʃ/.

Answer: (b)

註(a)：a computer fetches its first instruction from its ROM

註(b)：/kæ ʃ/ is the original pronunciation; /keɪʃ/ is widely heard in the IT world and elsewhere.

6. Which of the following is (are) true?

   (a) Virtual memory technique treats the main memory as a fully-set associative write-through cache.

   (b) Virtual address must be always larger than the physical address.

   (c) A TLB can be seen as the cache of a page table.

   (d) If an instruction TLB miss occurs, an instruction page fault is signaled.

Answer: (c)

註(a)：virtual memory treats the main memory as a fully-set associative write-back cache.

7. Which of the following is (are) true?

   (a) When a cache write miss occurs, the written data are only updated in the main memory. This is the write-around policy.

   (b) There is no cache coherency problem for the write-through cache since the data are written into the next level of memory.

   (c) When a cache write hit occurs, the written data are only updated in the cache. This is the write-back policy.

   (d) Cache data inconsistency appears in a write-back cache when an I/O master writes data into the memory block which is cached.

Answer: (a), (c), (d)

8. Registers in a processor are typically synthesized from D-FFs.

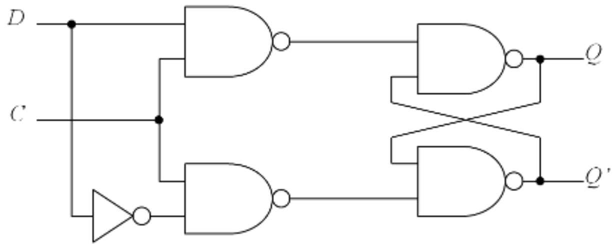
   (a) Show the schematic of a gated D-latch based on S-R latch.

   (b) Use the gated D-latch above to construct a D flip-flop.

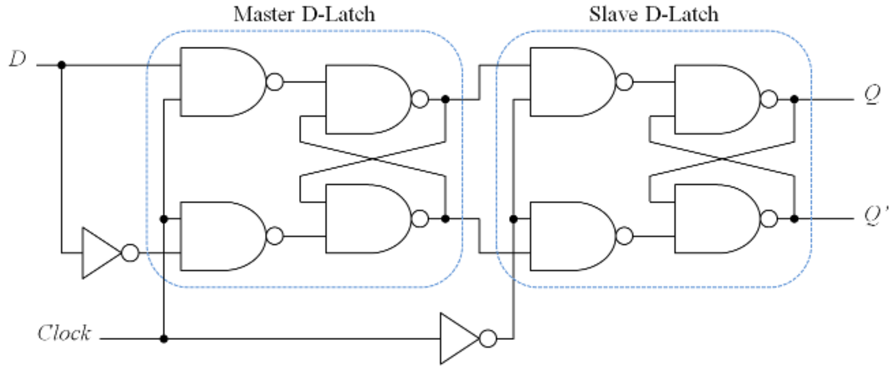
   (c) Show a 4-bit register based on the above D flip-flop.

Answer

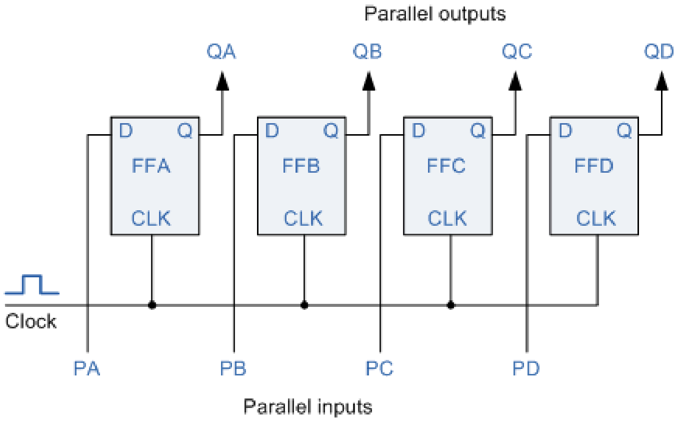
(a)



(b)


Master D-Latch
Slave D-Latch
D
Q
Q'
Clock

(c)


Parallel outputs
QA
QB
QC
QD
D
Q
FFA
CLK
D
Q
FFB
CLK
D
Q
FFC
CLK
D
Q
FFD
CLK
Clock
PA
PB
PC
PD
Parallel inputs

103 成大電通

1. Which of the following is (are) true?
   (a) For a fixed size cache memory, the smaller the line size is the smaller the spatial locality the cache has.
   (b) For a fixed size cache memory, the larger the line size is the smaller the temporal locality the cache has.
   (c) For a direct-mapped cache, no address tag is the same in the tag memory.
   (d) For a fully set associative cache, no address tag is the same in the tag memory.

Answer: (a), (d)

註(a)(b)： Temporal locality determine sensitivity to cache size and spatial locality determine sensitivity to line size.

2. Coding and ISA problem.
   (a) Write a C code for the execution of a 九九乘法表.
   (b) Convert the above C code into MIPS or ARM assembly or to RISC-like instructions.
   (c) Transform the above assembly code into a subroutine code.

Answer

(a)

```
int main()
{
 int i, j;
        for ( i = 1 ; i ≤ 9 ; i++ )
                {
                  for ( j = 1 ; j ≤ 9 ; j++ )
                  printf("%d*%d = %2d ", i, j, i*j);
                  printf("\n");
                }
return 0;
}
```

(b)(c)

Multiplication table:

```
                addi  $s0, $zero, 1          ; i = 1
loop1:          slti  $t0, $s0, 10           ;if (i < 10)
                beq   $t0, $zero, exit       ;if ( i ≥ 10) goto exit

                addi  $s1, $zero, 1          ; j = 1
loop2:          slti $t0, $s1, 10            ;if (j < 10)
                beq  $t0, $zero, label       ;if (j ≥ 10) goto label
```

```
          move $a0, $s0                 ; printf i
          li    $v0, 1
          syscall
          la    $a0, mul_str            ;printf *
          li    $v0, 4
          syscall
          move $a0, $s1                 ; printf j
          li    $v0, 1
          syscall
          la    $a0, equ_str            ;printf =
          li    $v0, 4
          syscall
          mul   $a0, $s0, $s1           ; printf i*j
          li    $v0, 1
          syscall
          addi $s1, $s1, 1              ;j++
          j     loop2
label:    la    $a0, ch_row             ;printf \n
          li    $v0, 4
          syscall
          addi $s0, $s0, 1              ;i++
          j     loop1
exit:     jr    $ra
```

103 成大資聯

1. Explain the following term in English. For each term, please use less than 100 words.
   (a) Power Wall
   (b) GPGPU
   (c) Restartable instruction
   (d) Write-allocate policy
   (e) NUMA

Answer

(a) Power wall: the trend of consuming exponentially increasing power with each factorial increase of operating frequency. The power wall is forcing a design toward simpler and more power-efficient processors on a chip.

(b) GPGPU: Using a GPU for general-purpose computation via a traditional graphics API and graphics pipeline.

(c) Restartable instruction: An instruction that can resume execution after an exception is resolved without the exception's affecting the result of the instruction.

(d) Write-allocate policy: When a write miss occurs, the corresponding block is fetched from memory and then the appropriate portion of the block is overwritten.

(e) NUMA: A type of single address space multiprocessor in which some memory accesses are much faster than others depending on which processor asks for which word.

2. Suppose we have developed new versions of a processor with the following characteristics.

Version	Voltage	Clock rate
Version 1	1.2V	810 MHz
Version 2	1V	1 GHz

(a) How much has the dynamic power been reduced if the capacitive load does not change?

(b) Assuming that the capacitive load of version 2 is 80% the capacitive load of version 1, find the voltage for version 2 if the dynamic power of version 2 is reduced by 20% from version 1.

Answer

(a) $\frac{1000\times 1^2}{810\times 1.2^2} = 0.86$, → dynamic power been reduced by 14%

(b) $\frac{1000\times 0.8\text{C}\times \text{V}^2}{810\times \text{C}\times 1.2^2} = 0.8$, → V = 1.08 Voltage

3. Translate the following C code to MIPS instructions. Assume that the variables c and d are assigned to registers $s0 and $s1, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

c = d - A[B[2]];

Answer

```
lw   $t0, 8($s7)
sll  $t0, t0, 2
add  $t0, $s6, $t0
lw   $t1, 0($t0)
sub  $s0, $s1, $t1
```

4. What decimal number does the following bit pattern represent if it is a floating point number? Use the IEEE 754 standard.

101011111011010000000000

Answer

$(-1)^1 \times (1.01101) \times 2^{95-127} = - 1.01101_2 \times 2^{-32} = - 1.40625_{10} \times 2^{-32}$

註：There are something wrong in this problem because only 24 bits in the given bit pattern.

5. Assume an instruction pipeline for a high-speed, load/store processor with the following instruction classes

ALU	ALUop	Rdst, Rsrc1, Rsrc2
ALUimmediate	ALUiop	Rdst, Rsrc1, imm
Load	MEMop	Rdst, n(Rsrc)
Store	Memop	n(Rsrc2), Rsrc1
Conditional branch	BRop	Rsrc1, Rsrc2, offset
Jumps	JMP	Rdst

Each instruction takes one machine word. The only memory-addressing mode supported is base register plus a signed offset. Conditional branches compare the two branch source operand values using the ALU. The branch target address is computed on a separate address generation adder contained in the control unit of the machine. Register file writes occur in the first half of a cycle and register file reads occur in the second half. The machine uses virtual memory address, with separate instruction and data TLBs. It also has a physical addressed (tagged) direct mapped L1 Icache and a physically addressed set associate L1 Dcache. (Accesses that miss in the L1 cache cause the instruction pipeline to stall.) The ALU used during the EX cycle is pipelined and takes one cycle to complete an addition/subtraction/logical operations and two cycles to complete a multiplication. The time taken by the key components (that already includes the pipeline register write time, the interconnect delay, and any necessary multiplexors) is as follows:

I or D TLB access	1 ns
Icache access	2 ns
Instruction decode	1 ns
Address adder	1 ns
Dcache access	2 ns
ALU pipe stage	2 ns
RegFile read/write	2 ns per 2 port access (e.g. a 2-read and 2-write port RegFile takes 2 ns, a 4- read-4 write port RegFile takes 4 ns, etc.)

Answer the following questions about your pipeline.

(a) Draw the shortest possible instruction pipeline (i.e. the pipeline with the fewest stages) while ensuring that there are no structural hazards. For your pipeline, give a name for each stage along with a short description of what activities occur during that stage.

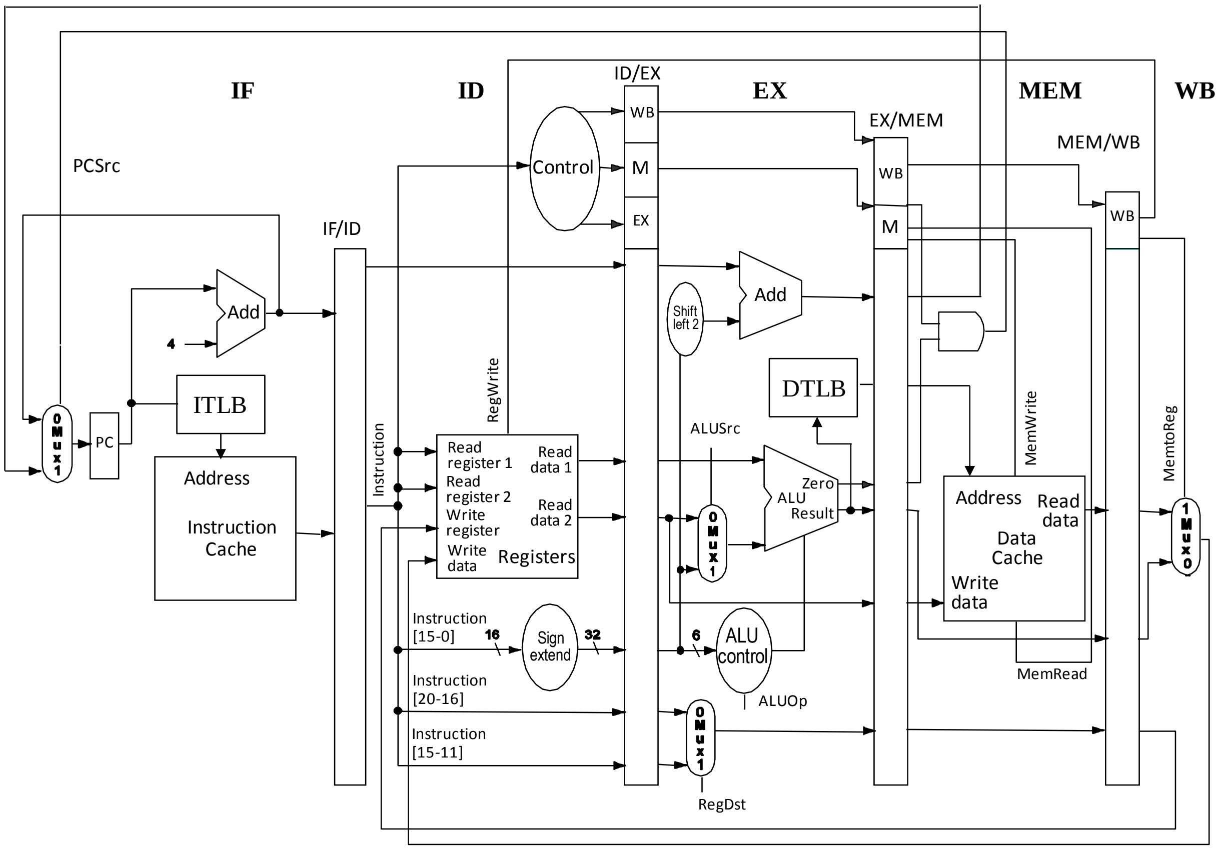
(b) What is its clock rate?

(c) Give a MIPS instruction example of two different data hazards that can be solved by forwarding (both data hazards should be different in that the forwarding is handled from different pipeline register stages). For each, explain which data are forwarded.

(d) Give a MIPS instruction example of two different data hazards that cannot be solved by forwarding. For each, indicate how many stalls are incurred before the hazard is resolved.

Answer

(a)



IF: Instruction Fetch

ID: Instruction Decode and Register File Read

EX: Execution or Address Calculation

MEM: Data Memory Access

WB: Write Back

(b) The longest stage time = 1ns + 2ns = 3ns (ITLB + Dcache or ALU + DTLB)

→ clock rate = 0.33 GHz

(c)

Example1 (EX hazard)	Example1 (MEM hazard)
add $1, $2, $3 sub $4, $1, $2	add $1, $2, $3 slt $5, $6, $7 sub $4, $2, $1
$1 is forwarded from MEM stage	$1 is forwarded from WB stage

(d)

Example (Load-use data hazard)	
lw $1, 12($2) add $2, $1, $3	After stall one clock cycle between lw and add the data of $1 will be forwarded from WB stage

103 中央資工

1. Which of the following statements are true about the assumption and definition of branch prediction?
   (a) Data that is being operated on has regularities.
   (b) Underlying algorithm has regularities.
   (c) Static branch prediction usually outperforms dynamic branch prediction.
   (d) The hypothesis of correlating branches is that the behavior of other branch instructions do not affect the prediction of current branch.
   (e) In general, an (m, n) correlating predictor records the last m branches to select between m history tables each with n-bit counters.

Answer: (a), (b)

註(e): In general, (m, n) predictor means record last *m* branches to select between 2^m history tables, each with *n*-bit counters

2. Suppose that FPSQR instructions are improved with speedup=10. FPSQR instructions are responsible for 20% of the execution time. What is the overall speedup?
   (a) 1.08 (b) 1.22 (c) 1.35 (d) 1.55 (e) 1.67

Answer: (b)

註: Overall speedup = 1 / (0.8 + 0.2 / 10) = 1.22

3. Compute the clock cycle per instruction (CPI) for the following instruction mix. The mix includes 22% loads, 11% stores, 49% R-format operations, 16% branches, and 2% jumps. The number of clock cycles for each instruction class is listed as follows. 5 cycles for loads, 4 cycles for stores, 4 cycles for R-format instructions, 3 cycles for branches, 3 cycles for Jumps.
   (a) 3.58 (b)3.76 (c) 4.04 (d) 4.28 (e) 4.52

Answer: (c)

註: CPI = 0.22 × 5 + 0.11 × 4 + 0.49 × 4 + 0.16 × 3 + 0.02 × 3 = 4.04

4. Which of the following statements are true?
   (a) RISC architecture usually needs more special purpose registers than CISC.
   (b) For a cache with write back strategy, read misses might result in writes.
   (c) The advantages of dynamic scheduling include memory latency hiding and resolving real dependence which is unknown at compile time.
   (d) Out-of-order completion might result in write after write (WAW) and write after read (WAR) hazards.
   (e) Distributed shared-memory scheme might result in non-uniform memory access time for multiprocessor machines.

Answer: (b), (d), (e)

註(b): when the victim block is dirty, it need to be write back to memory

註(c): not include memory latency hiding

5. What is the average memory access time for the memory with the following two-way set-associative cache? The hit time is 1.1 clock cycle. The miss rate is 0.04. And the miss penalty is 8 clock cycles.
   (a) 1.02 clock cycles (b) 1.12 clock cycles (c) 1.22 clock cycles (d) 1.32 clock cycles (e) 1.42 clock cycles

Answer: (e)

註: AMAT = 1.1 + 0.04 × 8 = 1.42

6. K is the 32-bit IEEE754 floating-point number of (1987/6). N is the number of bits equal to '1' in K. What is "N mod 5"?
   (a) 0 (b) 1 (c) 2 (d) 3 (e) 4

Answer: (a)

註: $(1987/6)_{10} = 101001011.0\overline{01}_2 = 1.01001011\,0\overline{01} \times 2^8$

→ K = 0 10000111 01001011001010101010101 → N =15

15 mod 5 = 0

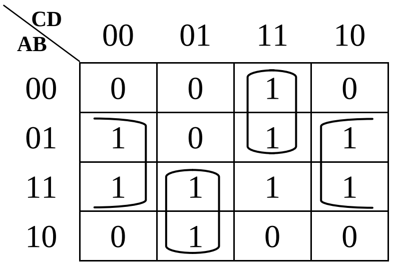
7. Which of the following statement(s) are correct?
   (a) Inserting L2 cache will not affect the miss rate of L1 cache.
   (b) The case of "TLB hit. Page Table hit. Cache miss" is possible.
   (c) One of the advantages of the superscalar processor over VLIW is the backward compatibility of the software.
   (d) In a pipelined system, forwarding can help to eliminate all the stalls resulting from data hazards.
   (e) The advantage of using microprogramming in designing the control is the execution speed.

Answer: (a), (b), (c)

8. Which of the following sum terms must be included in the sum-of-products simplification for (A'+B+C')(A+C+D')(A+B+C)(B+D)
   (a) AB (b) BD' (c) A'BD (d) AC'D (e) A'CD

Answer: (b), (d), (e)

註： f = (A'+B+C')(A+C+D')(A+B+C)(B+D) → $\overline{f}$=AB'C + A'C'D + A'B'C' + B'D' →



AB \ CD	00	01	11	10
00	0	0	1	0
01	1	0	1	1
11	1	1	1	1
10	0	1	0	0

Essential prime implicants → BD', A'CD, AC'D

9. A 32-bit cache memory address is decomposed into 3 fields as:

31 19	18 4	3 0
Tag	Set	Offset

Assume the cache is 2-way set associative. The total size (including tag and data bits but excluding other valid/dirty bits) of the cache memory is L KB(Kbytes). N=round(L) mod 5. Then N is equal to (a) 0 (b) 1 (c) 2 (d) 3 (e) 4

Answer: (d)

註： The total size = $(13 / 8 + 16) \times 2 \times 2^{15}$ = 1128 KB → L =1128

N = round(1128) mod 5 = 1128 mod 5 = 3

10. We want to evaluate the performances of two computers M1 and M2. M1 has a clock rate of 1GHz and M2 has a clock rate of 800MHz. The following shows the CPI of 3 instruction classes:

Instruction Class	CPI for M1	CPI for M2
A	3	2
B	4	4
C	4	3

There are 3 compilers: C1, C2, C3 that produce the instruction mixes as follows:

Instruction Class	C1	C2	C3
A	30%	25%	50%
B	50%	25%	30%
C	20%	50%	20%

Assume that any of the three compilers (C1, C2, C3) can be used in M1 and M2. We also assume that the numbers of instructions from C1 and C2 are the same for both M1 and M2 and equal to 1000, while the number of instructions from C3 is 1050 (for both M1 and M2). Which of the following combinations of compiler + computer will you choose?

(a) C1+M1 (b) C2+M1 (c) C3+M1 (d) C2+M2 (e) C3+M2

Answer: (c), (e)

註：

	M1	M2
C1	CPI = 0.3 × 3 + 0.5 × 4 + 0.2 × 4 = 3.7	CPI = 0.3 × 2 + 0.5 × 4 + 0.2 × 3 = 3.2
	ExTime = (1000 × 3.7) / 1G = 3.7 s	ExTime = (1000 × 3.2) / 800 M = 4 s
C2	CPI = 0.25 × 3 + 0.25 × 4 + 0.5 × 4 = 3.75	CPI = 0.25 × 2 + 0.25 × 4 + 0.5 × 3 = 3
	ExTime = (1000 × 3.75) / 1G = 3.75 s	ExTime = (1000 × 3) / 800 M = 3.75 s
C3	CPI = 0.5 × 3 + 0.3 × 4 + 0.2 × 4 = 3.5	CPI = 0.5 × 2 + 0.3 × 4 + 0.2 × 3 = 2.8
	ExTime = (1050 × 3.5) / 1G = 3.675 s	ExTime = (1050 × 2.8) / 800 M = 3.675 s

103 中正電機

1. Explain the following terms
   (a) Structure hazard and give its solutions
   (b) Traditional memory (ROM/RAM) vs. associative memory
   (c) RISC vs. CISC
   (d) Mainframe computer
   (e) Opcode vs. operand

Answer

(a) **Structure hazard :** When a planned instruction cannot execute in the proper clock cycle because the hardware does not support the combination of instructions that are set to execute.

Solution: Stall the pipeline for one clock cycle when the conflict is detected.

(b) **Traditional memory:** Traditional memory stores data at a unique address and can recall the data upon presentation of the complete unique address.

Associative memory: associative memory is also referred to as content-addressable memory. Associative memory can be directly accessed by the content rather than the physical address in the memory.

(c) The primary goal of **CISC** architecture is to complete a task in as few lines of assembly as possible.

RISC processors only use simple instructions that can be executed within one clock cycle.

(d) **Mainframes *computer*** *is a powerful and expensive computer capable of supporting hundreds, or even thousands, of users simultaneously.*

(e) **Opcode** is the portion of a machine language instruction that specifies the operation to be performed.

Operand is a memory location or a variable or any general purpose register. It stores data and you can perform operations on it.

2. The following formula is 64-bit version of IEEE754 standard.
   $N = (-1)^s \, 2^{E-1023} \, (1.M) \qquad 0 < E < 2047$
   It employs an 11-bit exponent E and a 52-bit mantissa M.
   (a) Find the representation (E and M) of N= -5.03.
   (b) Find the largest positive number.

Answer

(a) $-5.03_{10} = -101.00\overline{00011110101110000101} = -1.0100\overline{00011110101110000101} \times 2^2$

→E = 1025, M = 0100000111101011100001010001111010111000010100011110

(b) $1.1111111111111111111111111111111111111111111111111111 \times 2^{1023}$
$\approx 2 \times 10^{308}$

3. Calculate X × Y(X=101011, Y=101011) and list the operation process.
(a) By Booth's multiplication algorithm.
(b) By Modified Booth's multiplication algorithm.

Answer

(a) Booth's multiplication algorithm (2-bit)

Iteration	Step	Multiplicand	Product
0	initial values	101011	000000 101011 0
1	10→ – Multiplicand	101011	010101 101011 0
	Shift right product	101011	001010 110101 1
2	11→ none	101011	001010 110101 1
	Shift right product	101011	000101 011010 1
3	10→ + Multiplicand	101011	110000 011010 1
	Shift right product	101011	111000 001101 0
4	01→ – Multiplicand	101011	001101 001101 0
	Shift right product	101011	000110 100110 1
5	10→ + Multiplicand	101011	110001 100110 1
	Shift right product	101011	111000 110011 0
6	01→ – Multiplicand	101011	001101 110011 0
	Shift right product	101011	000110 111001 1

(b) Modified Booth's multiplication algorithm (3-bit)

Iteration	Step	Multiplicand	Product
0	initial values	101011	000000 101011 0
1	110→ – Multiplicand	101011	010101 101011 0
	Shift right product	101011	000101 011010 1
2	010→ – Multiplicand	101011	011010 011010 1
	Shift right product	101011	000110 100110 1
3	010→ – Multiplicand	101011	011011 100110 1
	Shift right product	101011	000110 111001 1

4. Fig.l and Fig.2 are the organization and instruction set of an 8 bits accumulator-based CPU. (AR: Address Register. PC: Program Counter. DR: Data Register, IR: Instruction Register. M: Memory, AC: Accumulator.)

(a) Give the Names of the micro-instructions of LD X for Fig.2 instruction set.

(b) Give the micro-operations of the instructions: ST X and ADD such as LD X.

The micro-operations of LD X:

AR:=PC, DR:=M(AR),

PC:=PC+1, IR:=DR(Opcode),

AR: DR(X),

DR:= M(AR),

AC:=DR.

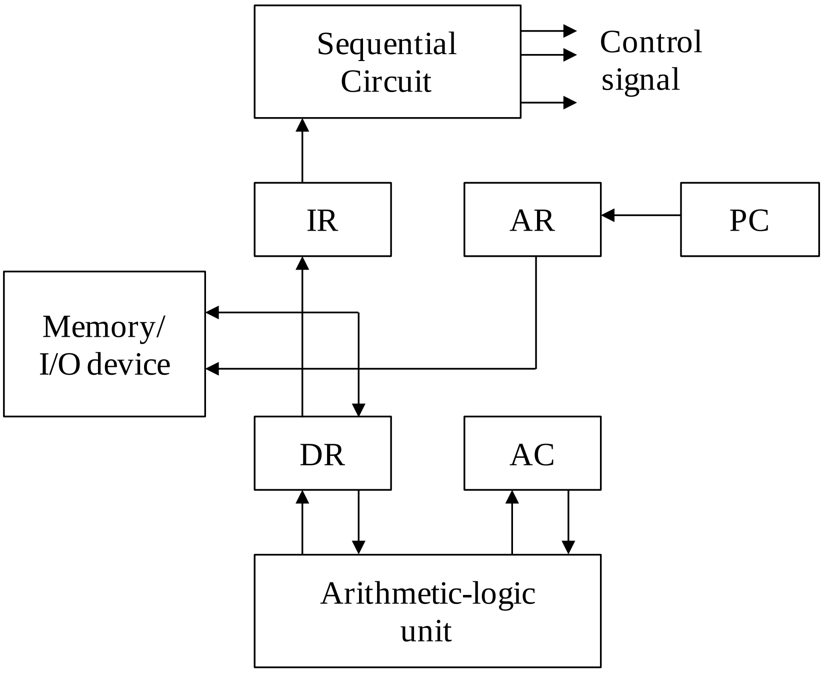


Fig. 1 An accumulator-based CPU

Instruction	HDL format	Comment
LD X	AC:= M(X)	Load X from the contents of M into AC
ST X	M(X):=AC	Store contents of AC in Memory
ADD	AC:=AC+DR	Add DR to AC
SUB	AC:=AC-DR	Subtract DR from AC
NOT	AC:= not AC	Complement contents of AC
MOV	DR:=AC	Copy contents of AC to DR

Fig.2 The instruction set for CPU of Fig. 1.

Answer

(a)

Micro-operation	Name of micro-operation	
AR:=PC, DR:=M(AR)	Instruction fetch	
PC:=PC+1, IR:=DR(Opcode)	Instruction decode	
AR: DR(X)	Transfer address field to AR	Execution
DR:= M(AR)	Transfer content of address to DR	
AC:=DR	Transfer content of DR to AC	

(b)

Instructions	ST X	ADD

Micro-operations	AR:= PC
	DR:= M(AR)
	PC:= PC + 1
	IR:= DR(Opcode)
	AR:= DR(X)
	AC:= DR
	M(AR):= DR
		AR:= PC
		DR:= M(AR)
		PC:= PC+1
		IR:= DR(Opcode)
		AC:= AC + DR

5. True or False

(a) CPU performance can be improved by increasing the clock rate and the number of clock cycles.

(b) Adding more CPU cores can improve the response time of each instruction.

(c) Adding more CPU cores can improve the throughput of a computer system.

(d) The power consumption of a CPU is proportional to the supply voltage.

(e) The power consumption of a CPU is proportional to the clock rate.

(f) After improving an aspect of computer, such as the floating-point multiplier, we can expect a proportional improvement in overall performance.

(g) MIPS (millions of instructions per second) is a better performance metric than CPU time.

(h) Registers are faster to access than memory.

Answer

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
False	False	True	False	True	False	False	True

6. Schedule the following codes for a dual-issue MIPS processor with at most four clock cycles. The processor can fetch one ALU/branch instruction and one load/store instruction at each clock cycle.

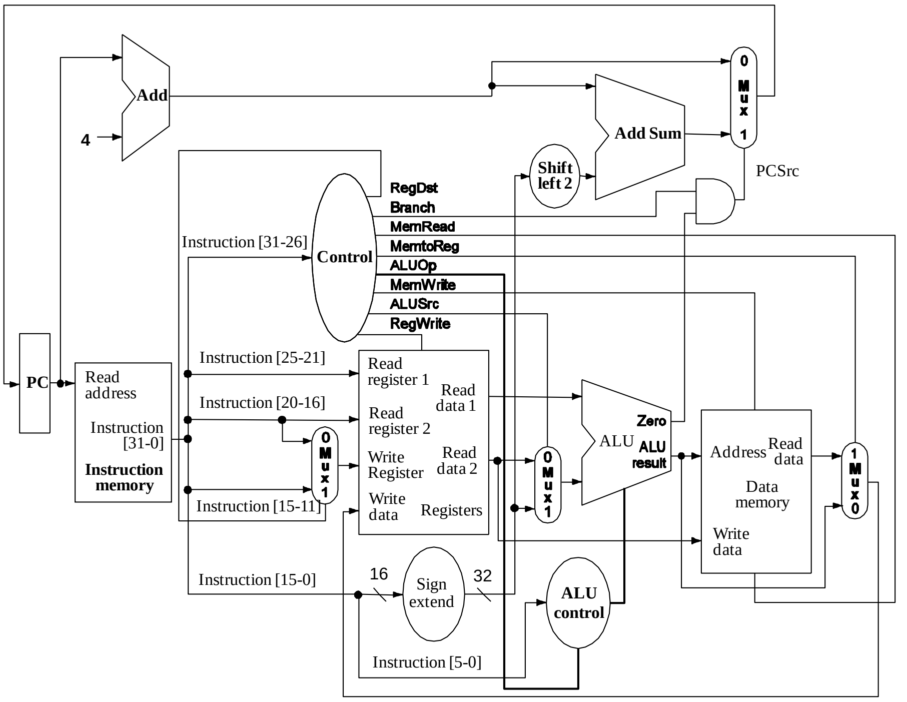
```
Loop:   lw      $t0, 0($s1)         # $t0=array element
        addu    $t0, $t0, $s2       # add scalar in $s2
        sw      $t0, 0($s1)         # store result
        addi    $s1, $s1, -4        # decrement pointer
        bne     $s1, $zero Loop     # branch $s1 !=0
```

	ALU/branch	Load/store	Cycle
Loop			1
			2
			3
			4

Answer

	ALU/branch	Load/store	Cycle
Loop		lw $t0, 0($s1)	1
	addi $s1, $s1, -4		2
	addu $t0, $t0, $s2		3
	bne $s1, $zero Loop	sw $t0, 4($s1)	4

7. Given the datapath and control signals of a MIPS CPU below, please specify the values of the eight control signals, "RegDst", "Branch", "MemRead", "MemtoReg", "ALUOp", "MemWrite", "ALUSrc", and "RegWrite", when fetching the instruction "sub $t0, $t1, $t2".



opcode	ALUOp	Operation	funct	ALU function	ALU control
lw	00	load word	XXXXXX	add	0010
sw	00	store word	XXXXXX	add	0010
beq	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
		subtract	100010	subtract	0110
		AND	100100	AND	0000
		OR	100101	OR	0001
		set-on-less-than	101010	set-on-less-than	0111

Answer

RegDst	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	0	0	0	01	0	0	1

8. Fill in "Machine Language", "High-level Language", and "Assembly Language" in the following blanks.

   __(a)__ → **Compiler** → __(b)__ → **Assembler** → __(c)__

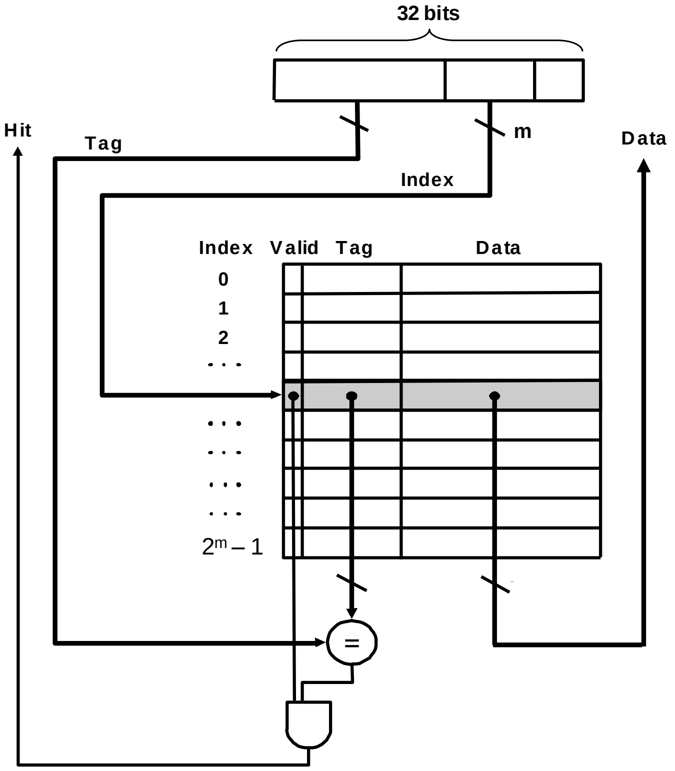
Answer

(a)	(b)	(c)
High-level Language	Assembly Language	Machine Language

9. A direct-mapped cache is designed based on 32-bit byte addresses. It contains 2^m blocks, and each block has 2^n words. Draw the architecture of the cache, and calculate the size of the cache?

Answer

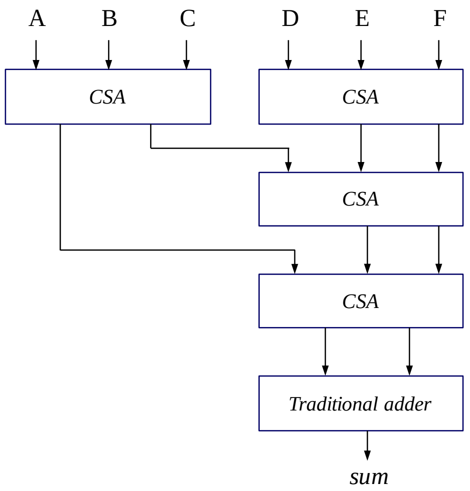
Cache size = $(1 + 32 – m – n – 2 + 32 \times 2^n) \times 2^m$ bits

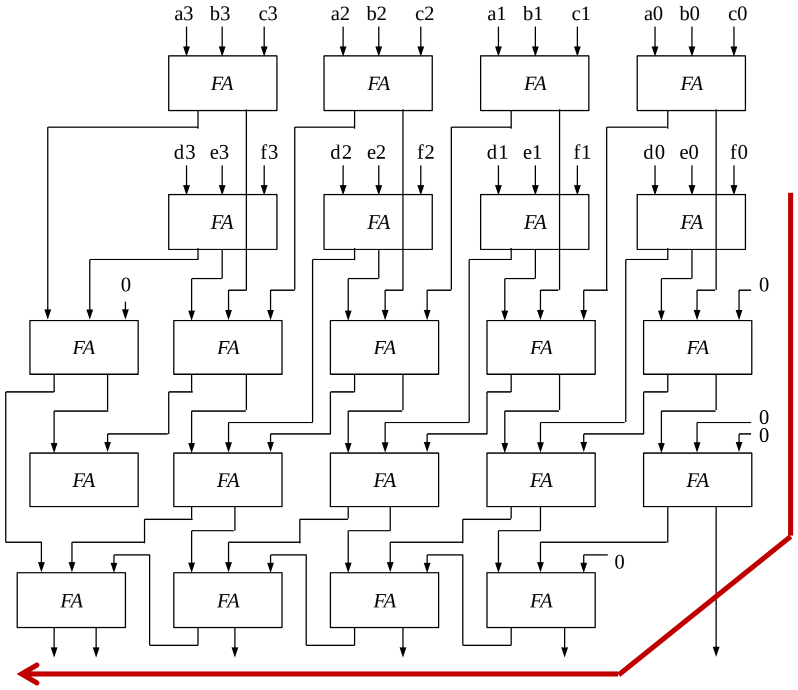


103 中正資工

1. If we want to design a carry-save adder to compute the addition of six 4-bit unsigned numbers: A, B, C, D, E, F with ONLY 1-bit full adders, and the delay time of the full adder is D_{FA}. Please determine the minimum delay time for this carry-save adder.

Answer: $7D_{FA}$





Critical Path

2. The following techniques have been developed to reduce cache miss penalty or miss rate: "critical word first and early restart", "pipelined cache access", and "hardware prefetching of instructions". Please briefly explain these techniques and how they work.

Answer

Critical word first and early restart: Don’t wait for full block to be loaded before restarting CPU.

Critical word first: Request the missed word first from memory and send it to the CPU as soon as it arrives; let the CPU continue execution while filling the rest of the words in the block.

Early restart: As soon as the requested word of the block arrives, send it to the CPU and let the CPU continue execution

Pipelined cache access:, Pipelining the cache access by inserting latches between modules in the cache can achieve a high bandwidth cache. Cache access time is divided into decoding delay, wordline to sense amplifier delay, and mux to data out delay. Using this technique, cache accesses can start before the previous access is completely done, resulting in high bandwidth and a high frequency cache.

Hardware prefetching of instructions: Typically, CPU fetches 2 blocks on miss: requested and next. Requested block goes in instruction cache, prefetched goes in instruction stream buffer.

When the next instruction is actually needed, the instruction can be accessed much more quickly from the instruction stream buffer than if it had to make a request from memory.

103 中山電機

1. Terminology Explanation
   (a) CISC (b) RISC (c) Reconfigurable Computing (d) Multi-core Processor

Answer

(a) **CISC** stands for complex instruction set computer and is the name given to processors that use a large number of complicated instructions, to try to do more work with each one.

(b) **RISC** stands for reduced instruction set computer and is the generic name given to processors that use a small number of simple instructions, to try to do less work with each instruction but execute them much faster.

(c) **Reconfigurable Computing** is a computer architecture combining some of the flexibility of software with the high performance of hardware by processing with very flexible high speed computing fabrics like field-programmable gate arrays (FPGAs).

(d) **Multi-core Processor** is a microprocessor containing multiple processors ("cores") in a single integrated circuit.

2. Suppose we are considering a change to an instruction set. The base machine is a load-store machine. Measurements of the load-store machine showing the instruction mix and clock cycle counts per instructions are given in the following table:

Instruction Type	Frequency	Clock Cycle Count
ALU Operations	40%	1
Loads	20%	2
Stores	15%	2
Branches	25%	2

Let's assume that 25% of the ALU operations directly use a loaded operand that is not used again. We propose adding ALU instructions that have one source operand in memory. These new register-memory instructions have a clock cycle count of 2. Suppose that the extended instruction set increases the clock cycle count for branches by 1, but it does not affect the clock cycle time. Would this change improve CPU performance? Explain your answer.

Answer

The frequency of register-memory ALU instructions = 0.4 × 0.25 = 0.1

The frequency of ALU instructions becomes 0.4 – 0.1 = 0.3

The frequency of Load instructions becomes 0.2 – 0.1 = 0.1

The CPI for branch instructions becomes 2 + 1 = 3

$\text{CPI}_{\text{old}} = 0.4 \times 1 + 0.2 \times 2 + 0.15 \times 2 + 0.25 \times 2 = 1.6$

$\text{CPI}_{\text{new}} = (0.3 \times 1 + 0.1 \times 2 + 0.1 \times 2 + 0.15 \times 2 + 0.25 \times 3) / 0.9 = 1.94$

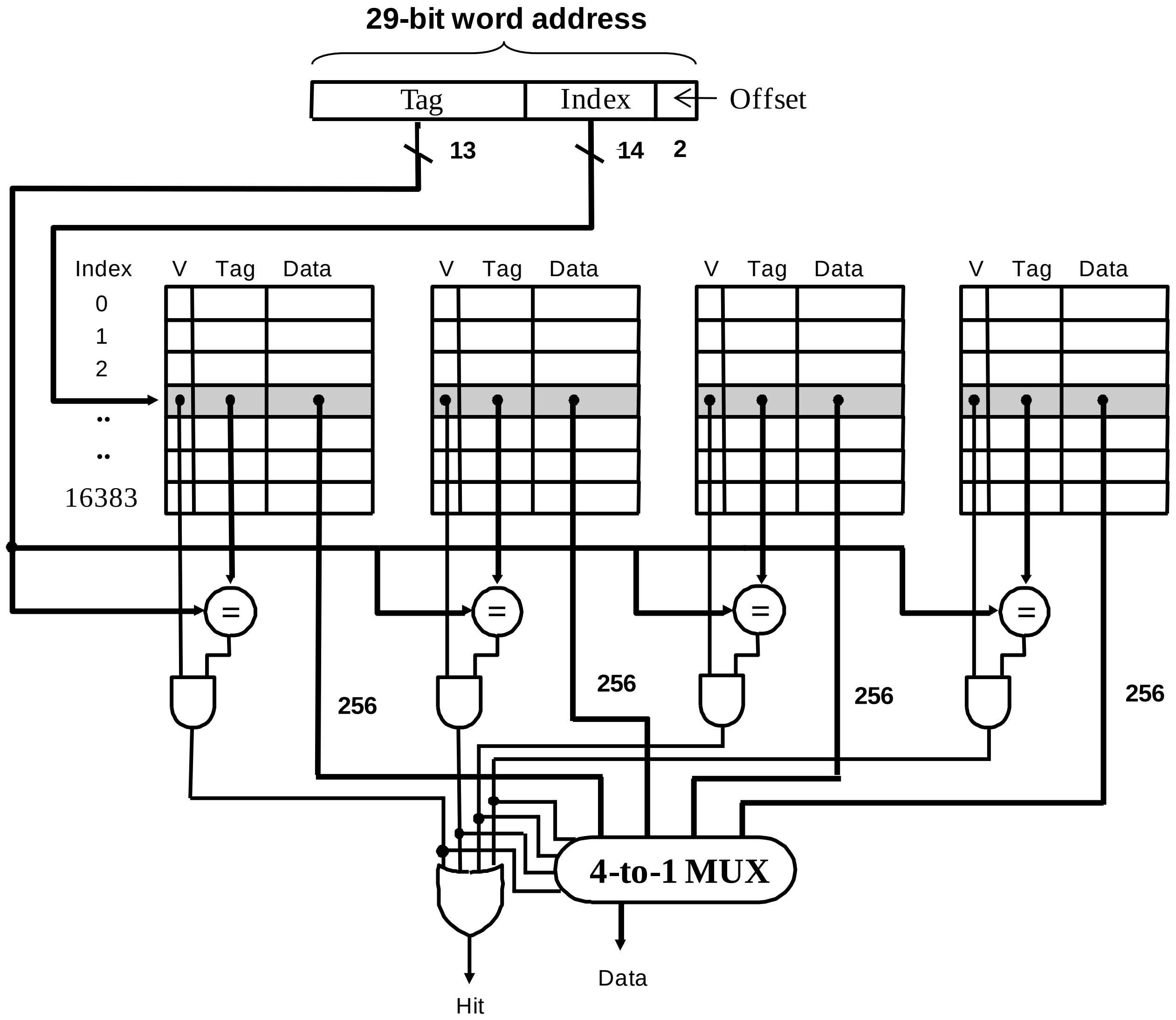
$\text{ExTime}_{\text{old}} = \text{IC} \times \text{CPI} \times \text{T} = 1.6 \times \text{IC} \times \text{T}$

$\text{ExTime}_{\text{new}} = 0.9\ \text{IC} \times \text{CPI} \times \text{T} = 0.9 \times 1.94 \times \text{IC} \times \text{T} = 1.75 \times \text{IC} \times \text{T}$

This change would not improve CPU performance

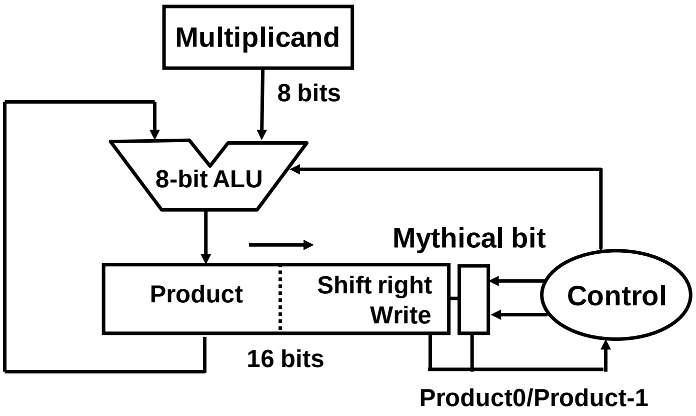
3. A set associative cache has a block size of four 64-bit words and a set size of 4. The cache can accommodate a total of 256K words. The main memory size that is cacheable is 512M × 64 bits. Design the cache structure and show how the processor's addresses are interpreted.

Answer



4. Design an 8-bits multiplier in *Booth's Algorithm.*

Answer



5. Use the following code fragment:

```
Loop: LW    R1, 0(R2)
      ADDI  R1, R1, #1
      SW    R1, 0(R2)
      ADDI  R2, R2, #4
      SUB   R4, R3, R2
      BNEZ  R4, Loop
```

Assume the initial value of R3 is R2+200. Use the five-stage instruction pipeline (IF, DEC, EXE, MEM, WB) and assume all memory accesses are one cycle operation. Furthermore, branches are resolved in MEM.

(a) Show the timing of this instruction sequence for the five-stage instruction pipeline with normal forwarding and bypassing hardware. Assume that branch is handled by predicting it has not taken. How many cycles does this loop take to execute?

(b) Assuming the five-stage instruction pipeline with a single-cycle delayed branch and normal forwarding and bypassing hardware, schedule the instructions in the loop including the branch- delay slot. You may reorder instructions and modify the individual instruction operands, but do not undertake other loop transformations that change the number of op-code of instructions in the loop. Show a pipeline timing diagram and compute the number of cycles needed to execute the entire loop.

Answer

(a)

Instructions	Clock cycle number													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LW R1, 0(R2)	F	D	X	M	W									
ADDI R1, R1, #1		F	D	D	X	M	W							
SW R1, 0(R2)				F	D	X	M	W						
ADDI R2, R2, #4					F	D	X	M	W					
SUB R4, R3, R2						F	D	X	M	W				
BNEZ R4, Loop							F	D	D	X	M	W		
Flushed instruction									F	D	X	M	W	
LW R1, 0(R2)										F	D	X	M	W

The total number of iterations is 200 / 4 = 50 cycles

There are 2 RAW hazards (2 stalls) and a flush after the branch since the branch is taken.

For the first 49 iterations, it takes 9 cycles between loop instances.

The last loop takes 12 cycles since the latency cannot be overlapped with additional loop instances. So, the total number of cycles is 49 × 9 + 12 = 453.

(b)

```
LW    R1, 0(R2)
ADDI  R2, R2, #4
SUB   R3, R3, R2
ADDI  R1, R1, #1
BNEZ  R3, Loop
SW    R1, -4(R2)
```

Instructions	Clock cycle number									
	1	2	3	4	5	6	7	8	9	10
LW R1, 0(R2)	F	D	X	M	W					
ADDI R2, R2, #4		F	D	X	M	W				
SUB R3, R3, R2			F	D	X	M	W			
ADDI R1, R1, #1				F	D	X	M	W		
BNEZ R3, Loop					F	D	X	M	W	
SW R1, -4(R2)						F	D	X	M	W

The total number of iterations is 200 / 4 = 50 cycles

For the first 49 iterations, it takes 6 cycles between loop instances.

The last loop takes 12 cycles since the latency cannot be overlapped with additional loop instances. So, the total number of cycles is 49 × 6 + 10 = 304.

103 中山資工

1. Consider two different implementations of the same instruction set architecture. There are four classes of instructions. A, B, C, and D. The clock rate and CPI of each implementation are given in the following table.

		Clock Rate	CPI Class A	CPI Class B	CPI Class C	CPI Class D
(a)	P1	2.5 GHz	1	2	3	3
	P2	3 GHz	2	2	2	2
(b)	P1	2.5 GHz	2	1.5	2	1
	P2	3 GHz	1	2	1	1

(a) Given a program with 10^6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?

(b) What is the global CPI for each implementation?

Answer

(a) ExTime for P1 = $\frac{10^6 \times (0.1\times1 + 0.2\times2 + 0.5\times3 + 0.2\times3)}{2.5\times10^9}$ = 1040 μs

ExTime for P2 = $\frac{10^6 \times 2}{3\times10^9}$ = 667 μs

P2 is faster than P1

(b) CPI for P1 = 0.1×2 + 0.2×1.5 + 0.5×2 + 0.2×1 = 1.7

CPI for P1 = 0.1×1 + 0.2×2 + 0.5×1 + 0.2×1 = 1.2

2. (a) $(520)_{10}$ = ($\quad\quad$ $)_8$ = ($\quad\quad$ $)_{16}$.

(b) Convert the decimal number -320 into signed 10-bit binary number in the 2's complement representation.

(c) Assume that signed numbers are stored in 10-bit words in 2's complement representation. What is the result of $(256)_{10}$ + $(257)_{10}$?

(d) Show the IEEE 754 binary representation of the number $(-0.75)_{10}$ in single precision.

Answer

(a) $(520)_{10} = (1010)_8 = (208)_{16}$

(b) $-320_{10} = 11011000000_2$

(c) The 10-bit 2's complement number can ranged from -512 to 511; therefore, $(256)_{10}$ + $(257)_{10}$ will result in **overflow**.

(d) $(-0.75)_{10} = (-0.11)_2 = -1.1 \times 2^{-1}$

→ 1 01111110 10000000000000000000000

3. True or False Questions: (12%)
   (a) A multi-core processor consists of multiple chips, each containing a processor.
   (b) A server is a computer composed of hundreds to thousands of processors and terabytes of memory.
   (c) Access time to random access memory (RAM) is longer than access time to hard disk.
   (d) Assembly language consists of commands that processors understand.
   (e) The application software is software/programs developed by the users.
   (f) NaN in IEEE 754 can be the result of dividing 0 by 0 (0/0).

Answer

(a)	(b)	(c)	(d)	(e)	(f)
False	False	False	False	False	True

4. Consider the segment of MIPS assembler code below.

```
lw    $v1, 0($a0)
add   $v2, $v2, $v1
sw    $v2, 0($a1)
addi  $a0, $a0, 1
```

   (a) How many times is instruction memory accessed? How many times is data memory accessed? (Count only accesses to memory, not registers.)
   (b) What are the addressing modes used in the four instructions?

Answer

(a) Instruction memory accessed: 4 times

Data memory accessed: 2 times

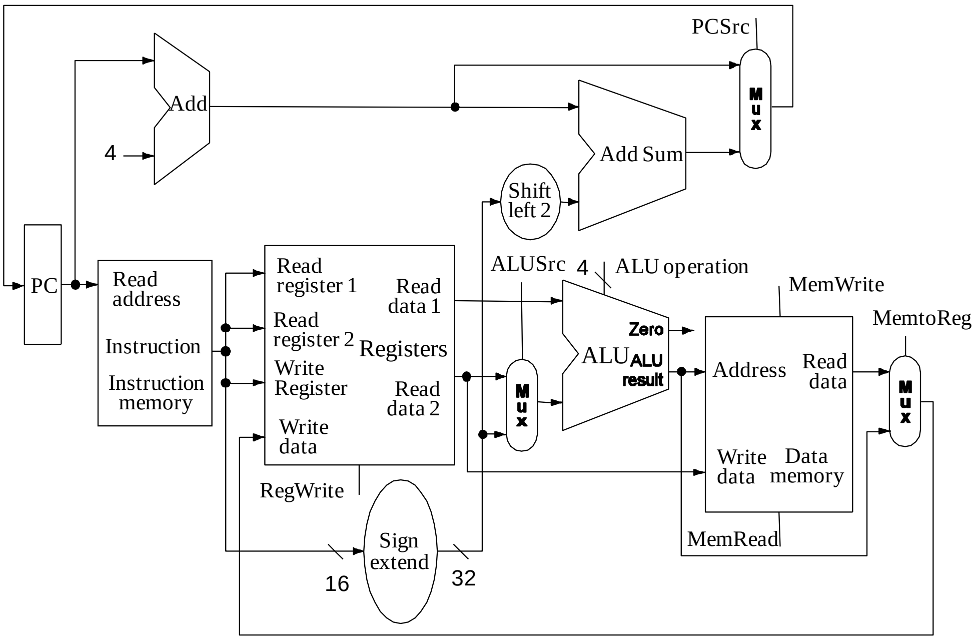
(b)

Instruction	Addressing mode
lw $v1, 0($a0)	Base or displacement
add $v2, $v2, $v1	Register
sw $v2, 0($a1)	Base or displacement
addi $a0, $a0, 1	Immediate

5. The latency of a block of digital logic is the time needed to do the work in the block. Assume that logic blocks needed to implement a processor's datapath have the following latencies:

I-Mem	Add	Mux	ALU	Reg's	D-Mem	Sign-Extend	Shift-Left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps

Consider the following datapath using the above logic blocks.



(a) What is the clock cycle time if the only types of instructions we need to support are ALU instructions (ADD, AND, etc.)?

(b) What is the clock cycle time if we only have to support LW (load word) instructions?

(c) What is the clock cycle time if we must support ALU, branch, load/store instructions? (Hint: You need to find the overall latencies of the blocks in the longest path required to execute an instruction.)

Answer

Instruction	longest-path latency for instructions
ALU	200 + 90 + 20 + 90 + 20 + 90 = 510 ps
LW	200 + 90 + 90 + 250 + 20 + 90 = 740 ps
SW	200 + 90 + 90 + 250 = 630 ps
Branch	200 + 90 + 20 + 90 + 20 = 420 ps

(a) The clock cycle time is 510 ps

(b) The clock cycle time is 740 ps

(c) The clock cycle time is 740 ps

6. Let the above datapath be implemented with pipelining. Assume that individual stages of the datapath (Instruction Fetch, Instruction Decode, Execution, Memory access. Write Back) have the following latencies:

IF	ID	EXE	MEM	WB
250ps	350ps	150ps	300ps	200ps

(a) What is the clock cycle time in a pipelined and non-pipelined processor?

(b) What is the total latency of an LW instruction in a pipelined and non-pipelined processor?

Answer

(a)	Clock cycle time for pipelined processor	Clock cycle timefor non-pipelined processor
	350 ps	250 + 350 + 150 +300 + 200 =1250 ps
(b)	LW latency in a pipelined processor	LW latency in non-pipelined processor
	350 × 5 = 1750 ps	250 + 350 + 150 +300 + 200 =1250 ps

7. Assume that you are using a standard 5-stage pipelined processor (as shown in Problem 6) to execute the following MIPS instructions:

```
lw    $1, 40($6)
add   $6, $2, $2
sw    $6, 50($1)
```

(a) Indicate dependences and their type.

(b) Assume there is no forwarding in this pipelined processor. Indicate hazards and add NOP instructions to eliminate them.

Answer

(a) For register \$6, there are WAR dependence between instruction pair (lw, add)
For register \$1, there are RAW dependence between instruction pair (lw, sw)
For register \$6, there are RAW dependence between instruction pair (add, sw)

(b) There are data hazards in instruction pairs (lw, sw) and (add, sw).

```
lw    $1, 40($6)
add   $6, $2, $2
NOP
NOP
sw    $6, 50($1)
```

8. Cache system

(a) Consider a direct-mapped cache with 16KB of data and 16-byte blocks, assuming a 32-bit address. What is the length of the tag field? How many total bits are required, assuming a valid bit is used?

(b) Assume an instruction cache miss rate for gcc of 2% and a data cache miss rate of 4%. If a machine has a CPI of 2 without any memory stalls and the miss penalty is 40 cycles for all misses, determine how much faster a machine would run with a perfect cache that never missed. Assume 36% of instructions are loads/stores.

Answer

(a) The length of index field = $\log_2(16KB/16B) = 10$.

The length of offset field = $\log_2(16B) = 4$

The length of tag field = $32 - 10 - 4 = 18$

Total bits = $(1 + 18 + 16 \times 8) \times 1K = 147K$ bits

(b) $CPI_{effective} = 2 + 1 \times 0.02 \times 40 + 0.36 \times 0.04 \times 40 = 3.376$

Machine with perfect cache is 3.376 / 2 = 1.688 times faster.

9. Briefly explain the following terms:

(a) Page fault

(b) Direct memory access (DMA)

Answer

(a) Page fault: an even that occurs when an accessed page is not present in main memory.

(b) DMA: a mechanism that provides a device controller with the ability to transfer data directly to or from the memory without involving the processor.

103 中興電機

1. A DVD driver works in the Constant Linear Velocity (CLV) mode. The read head must interact with the concentric circles at a constant rate, whether it is accessing data from the inner or outermost portions of the disk. This is affected by varying the rotation speed of the disk, from 1800 revolutions per minute (RPM) at the center, to 600 RPM at the outside. Assume that the DVD drive reads 2MB of user data per second.
   (a) How many bytes can the center circle store?
   (b) How many bytes can the outside circle store?

Answer

Bytes on center circle = 2 MB/seconds × 1/1800 minutes/rev × 60 seconds/minutes = 66.67 KB

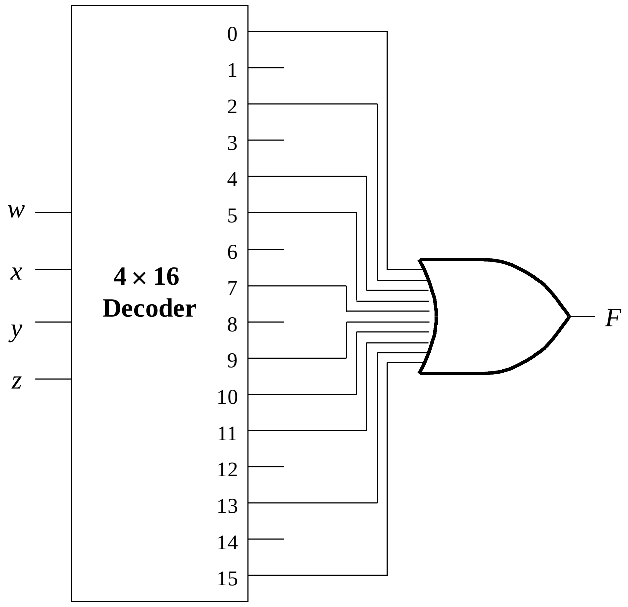
Bytes on outside circle = 2 MB/seconds × 1/600 minutes/rev × 60 seconds/minutes = 200 KB

2. Please implement of the following Boolean expression with a 4 × 16 decoder and an OR gate, and you can use a rectangle as a 4 × 16 decoder.

$$F(w, x, y, z) = \prod(l, 3, 6, 8, 12, 14)$$

Answer

$F(w, x, y, z) = \prod(l, 3, 6, 8, 12,14) = \Sigma(0, 2, 4, 5, 7, 9, 10, 11, 13, 15)$



3. The following C codes are compiled into the corresponding MIPS assembly codes. Assume that the two parameters *array* and *size* are found in the registers $a0 and $a1, and that i is allocated to register $t0.

C codes:

```
clearl(int array[], int size)
{
    int r,
        for (i=0; i < size; i+=1)
            array[i]=0;
}
```

MIPS assembly codes:

```
         move $t0, OP1
loop1:   sll  $t1, $t0, 2
         add  $t2, $a0, $t1
         sw   $zero, 0(OP2)
         addi $t0, OP3, l
         slt  $t3, $t0, $a1
         bne  OP4, $zero, loop1
```

Please determine proper values for the operands (OP1, OP2, OPS, OP4). Copy the following table (Table1) to your answer sheet and fill in the operand values.

Table 1

Operand	Value
OP1	
OP2	
OP3	
OP4	

Answer

Operand	Value
OP1	$zero
OP2	$t2
OP3	$t0
OP4	$t3

4. Assume that the miss rate of an instruction cache is 3% and the miss rate of the data cache is 6%. If a processor has a CPI of 4 without any memory stalls and the miss penalty is 100 cycles for all misses. Assume the frequency of all loads and stores is 30%. How much faster a processor will run with a perfect cache that never missed.

Answer

$CPI_{effective} = 4 + 1 \times 0.03 \times 100 + 0.3 \times 0.06 \times 100 = 8.8$

Processor with perfect cache is 8.8 / 4 = 2.2 times faster than which without perfect cache

5. For each code sequence below, state whether it must stall, can avoid stalls using only forwarding, or can execute without stalling or forwarding.

Sequence 1	Sequence 2	Sequence 3
lw $t0, 0($t0) add $t1, $t0, $t0	add $t1, $t0, $t0 addi $t2, $t0, #5 addi $t4, $t1, #5	addi $t1, $t0, #l addi $t2, $t0, #2 addi $t3, $t0, #3 addi $t4, $t0, #4 addi $t5, $t0, #5

Answer

Sequence 1: Stall on the LW result.

Sequence 2: Bypass the first ADD result written into $t1.

Sequence 3: No stall or bypass required.

6. We examine how pipelining affects the clock cycle time of the processor. Assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
300ps	400ps	350ps	500ps	100ps

(a) What is the clock cycle time in a pipelined and nonpipelined processor?

(b) What is the total latency of a lw instruction in a pipelined and nonpipelined processor?

(c) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

(d) Assume that instructions executed by the processor are broken down as follows:

ALU	beq	lw	sw
50%	25%	15%	10%

Assume there are no stalls or hazards, what is the utilization (% of cycles used) of the data memory?

(e) Repeat (d), assume there are no stalls or hazards, what is the utilization (% of cycles used) of the write-register port of the "Registers" unit?

Answer

(a) Clock cycle time for pipelined processor = 500 ps

Clock cycle time for nonpipelined processor = 300 + 400 + 350 +500 + 100 =1650 ps

(b) Latency of a lw instruction in the pipelined processor = 500 × 5 = 2500 ps

Latency of a lw instruction in the nonpipelined processor = 1650 ps

(c) MEM stage should be split

The new clock cycle time of the processor is 400 ps

(d) 15% + 10% = 25%

(e) 50% + 15% = 65%

7. Consider single precision IEEE 754 floating point numbers, as indicated below:

31	30 23	22 0
S	Exponent	Significand
1 bits	8 bits	23 bits

Given the following bit-encoding for the floating point numbers A and B:

A = 0100 0110 1101 1000 0000 0000 0000 0000 and

B = 1011 1110 1110 0000 0000 0000 0000 0000

Compute in BINARY and show the floating point encoding for the following in BINARY:

(a) A + B =

(b) A × B =

Answer

(a) A = 0100 0110 1101 1000 0000 0000 0000 0000 → A = +1.1011 × 2^{14}

B = 1011 1110 1110 0000 0000 0000 0000 0000 → B = -1.11 × 2^{-2}

Shift B right 16 bits → B = -0.000000000000000111 × 2^{14}

$$\begin{array}{r} 1.10110000000000000000000 \\ \underline{-\ 0.00000000000000011100000} \\ =1.10101111111111100100000 \end{array}$$

→ A + B = 1.10101111111111100100000 × 2^{14}

After rounding = 1.10101111111111100100000 × 2^{14}

Floating point encoding → 0 10001101 10101111111111100100000

(b) Exponent = 14 + (-2) = 12

$$\begin{array}{r} 1.10110000000000000000000 \\ \underline{\times\ \ 1.11000000000000000000000} \\ =1.00110100000000000000000 \end{array}$$

→ A × B = 1.00110100000000000000000 × 2^{12}

After rounding → 1.00110100000000000000000 × 2^{12}

Setting sign → -1.00110100000000000000000 × 2^{12}

Floating point encoding →1 10001011 00110100000000000000000

103 中興資工

1. Suppose we have two implementations of the same instruction set architecture, Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much? Please show your calculation steps.

Answer

Instruction time for Computer A = 2 × 250 = 500 ps

Instruction time for Computer B = 1.2 × 500 = 600 ps

Computer A is faster than computer B by 600/500 = 1.2 times faster

2. Suppose we developed a new, simpler processor that has 85% of the capacitive load of the more complex older processor. Furthermore, assume that it has adjustable voltage so that it can reduce voltage 15% compared to processor B, which results in a 15% shrink in frequency. What is the impact on dynamic power?

Answer

$\text{Power}_{\text{new}}$ / $\text{Power}_{\text{old}}$ = ((Capacitive load × 0.85) × $(\text{Voltage} \times 0.85)^2$ × (Frequency switched × 0.85)) / Capacitive Load × Voltage^2 × Frequency switched → the power ratio is $0.85^4 = 0.52$

Hence, the new processor uses about half the power of the old processor

3. C has many statements for decisions and loops, while ARM has few. Which of the following do or do not explain this imbalance? Why?
   (1) More decision statements make code easier to read and understand.
   (2) More decision statements simplify the task of the underlying layer that is responsible for execution.
   (3) Few decision statements mean fewer lines of code, which generally results in the execution of fewer operations.
   (4) Few decisions statements mean few lines of code, which generally reduces coding time.

Answer: (1), (2), (3), (4)

4. Subtract 6_{ten} from 7_{ten} in binary.

Answer

7 – 6 = 7 + (6 two's complement representation)

$$\begin{array}{rl}
 & 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0111_2 = 7_{10} \\
+ & 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1010_2 = 6_{10} \\
\hline
= & 000\ 0000\ 0000\ 00000\ 0000\ 0000\ 0000\ 0001_2 = 1_{10}
\end{array}$$

5. Figure 1 shows the data-path for the memory instructions and the R-type instructions. Which of the following is correct for a load instruction?
   (1) MemtoReg should be set to cause the data from memory to be sent to the register file,
   (2) MemtoReg should be set to cause the correct register destination to be sent to the register file.
   (3) We do not care about the setting of MemtoReg for loads.

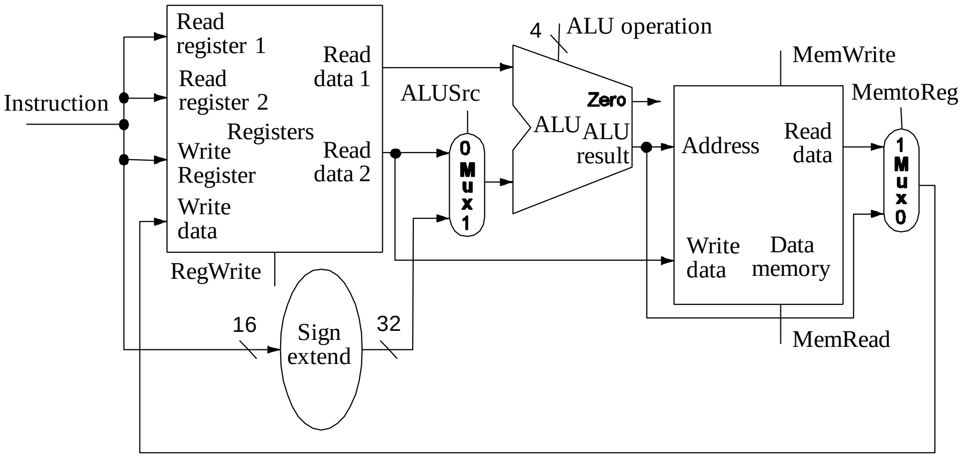


Figure 1. The datapath for the memory instructions and the R-type instructions

Answer: (1)

6. A group of students were debating the efficiency of the five-stage pipeline when one student pointed out that not all instructions are active in every stage of the pipeline. After deciding to ignore the effects of hazards, they made the following five statements. Which ones are correct?
   (1) After jumps, branches, and ALU instructions to take fewer stages than the five required by the load instruction will increase pipeline performance under all circumstances.
   (2) Trying to allow some instructions to take fewer cycles does help, since the throughput is determined by the clock cycle; the number of pipe stages per instruction affects both latency and thus throughput.
   (3) You cannot make ALU instructions take fewer cycles because of the write-back of the result, but branches and jumps can take fewer cycles, so there is some opportunity for improvement.
   (4) Instead of trying to make instructions take fewer cycles, we should explore making the pipeline longer, so that instructions take more cycles, but the cycles are shorter. This could improve performance.

Answer: (4)

103 台科大電子

1. Assume that a MIPS processor with a five-stage pipeline.
   (a) What is a five-stage pipeline?
   (b) What are the advantages of the pipeline?

Answer

(a) In MIPS five stages pipeline, datapath is divided into the following stages:
IF: instruction fetch
ID: instruction decode and register read
ALU: ALU execution
MEM: data memory read or write
WB: write result back into a register
Instructions move along the datapath, one stage at a time, through all stages.

(b) The advantages of the pipeline can overlap the execution of multiple instructions (increasing instruction throughput) to improve performance.

2. Please explain the following terms:
   (a) Delay branch
   (b) Dynamic branch prediction
   (c) Superscalar
   (d) TLB (translation-lookaside buffer)
   (e) Write back policy [hint: cache and memory]
   (f) Conflict miss
   (g) Set associative mapping

Answer

(a) Delay branch: a type of branch where the instruction immediately following the branch is always executed independent of whether the branch condition is true or false.

(b) Dynamic branch prediction: prediction of branches at runtime using runtime information.

(c) Superscalar: an advanced pipelining technique that enables the processor to execute more than one instruction per clock cycle by selecting them during execution.

(d) TLB: a cache that keeps track of recently used address mappings to try to avoid an access to the page table.

(e) Write back policy: a scheme that handles writes by updating values only to the block in the cache, then writing the modified block to the lower level of the hierarchy when the block is replaced.

(f) Conflict miss: A cache miss that occurs in a set-associative or direct mapped cache when multiple blocks compete for the same set and that are eliminated in a fully associative cache of the same size.

(g) Set associative mapping: a mapping in a cache that has a fixed number of locations (at least two) where each block can be placed.

3. Please answer the following questions:
   (a) What is data hazard?
   (b) How to handle data hazard in a pipeline?

Answer

(a) When a planned instruction cannot execute in the proper clock cycle because data that is needed to execute the instruction is not yet available.

(b) Insert nop instruction by compiler;
Reorder code sequence by compiler;
Forwarding by hardware.

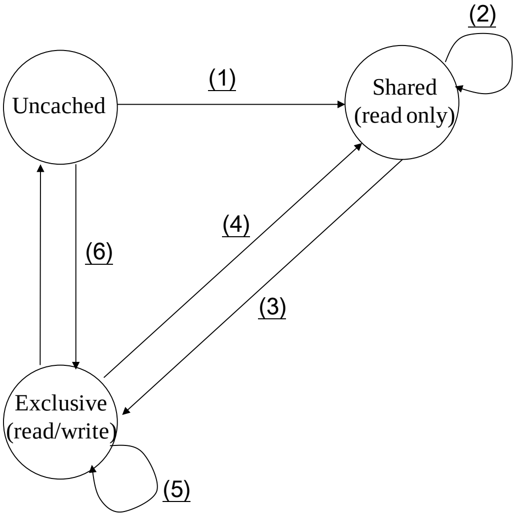
4. How to reduce cache miss? You will get more credits by giving methods as many as possible.

Answer

Increase block size to reduce compulsory misses;
Increase cache size to reduce capacity misses;
Increase associativity to reduce conflict misses;
Use more sophisticated replacement policy to reduce conflict misses.

103 台科大資工

1. Considering the directory protocol for distributed shared memory, please associate proper statements to the following state transition diagram for the directory. Note that P in the following statements means the requesting processor number.



A) Write Miss:
send Fetch/Invalidate;
send Data Value Reply msg to remote cache;
Sharers = {P};

B) Write Miss:
send Invalidate to Sharers;
then Sharers = {P};
send Data Value Reply msg

C) Write Miss:
Sharers = {P};
send Data Value Reply msg

D) Read miss:
Sharers += {P};
send Data Value Reply

E) Read miss:
Sharers = {P}
send Data Value Reply

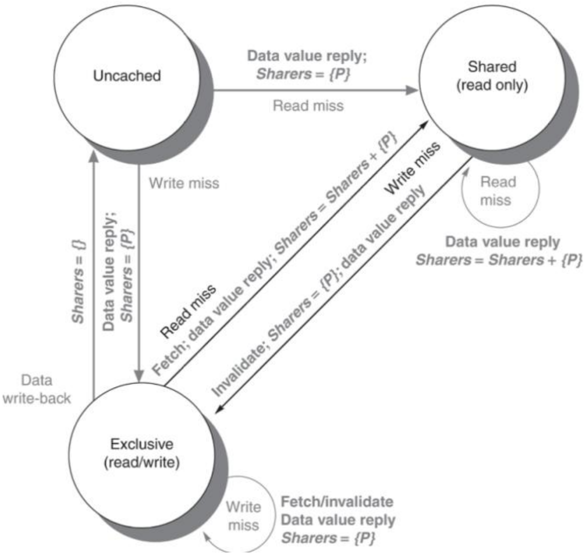
F) Read miss:
Sharers+={P};
send Fetch;
send Data Value Reply msg to remote cache (Write back block)

G) Data Write Back:
Sharers = {} (Write back block)

Answer

(1)	(2)	(3)	(4)	(5)	(6)
E	D	B	F	A	C

註: Directory-based Cache Coherence Protocols



2. Suppose we summarize the cache optimization techniques with "+" meaning that the technique improves the factor,"-" meaning it hurts that factor, and blank meaning it has no impact. For example, items (1), (2), (3), and (4) are all blank. Given the following table, please identify which items are "+" and which items are "-" (in an increasing order). (Note that give the correct/incorrect answer for an item, get/lose 1 point; otherwise, no point. The maximum points got are 16; the minimum points got are 0.)

Technique	Hit time	Bandwidth	Miss penalty	Miss rate
Banked caches	(1)	(12)	(23)	(34)
Compiler techniques to reduce cache misses	(2)	(13)	(24)	(35)
Compiler-controlled prefetching	(3)	(14)	(25)	(36)
Critical word first and early restart	(4)	(15)	(26)	(37)
Hardware prefetching of instructions and data	(5)	(16)	(27)	(38)
Merging write buffer	(6)	(17)	(28)	(39)
Nonblocking caches	(7)	(18)	(29)	(40)
Pipelined cache access	(8)	(19)	(30)	(41)
Small and simple cache	(9)	(20)	(31)	(42)
Trace caches	(10)	(21)	(32)	(43)
Way-prediction cache	(11)	(22)	(33)	(44)

Answer

Technique	Hit time	Bandwidth	Miss penalty	Miss rate
Banked caches		+		
Compiler techniques to reduce cache misses				+
Compiler-controlled prefetching			+	+
Critical word first and early restart			+	
Hardware prefetching of instructions and data			+	+
Merging write buffer			+	
Nonblocking caches		+	+	
Pipelined cache access	-	+		
Small and simple cache	+			-
Trace caches	+			
Way-prediction cache	+			

103 台師大資工

1. Consider the following code sequence.

```
SUB  R1, R4, R3        ; R1 ← R4 – R3
ADD  R5, R1, R6        ; R5 ←R1 + R6
LW   R8, 12(R6)        ; R8 ← MEM[12 + R6]
ADD  R9, R5, R8        ; R9 ← R5 + R8
SW   R9, 16(R6)        ; MEM[16 + R6] ← R9
```

Suppose the code sequence is executed in a five-stage (IF, ID, EX, MEM and WB) MIPS pipeline processor with hazard detection and data forwarding units. Assume the processor includes separate instruction and data memories so that the structural hazard for memory references can be avoided.

(a) Identify all the data hazards which can be solved by forwarding.

(b) Identify all the data hazards which can not be solved by forwarding.

(c) Determine the total number of clock cycles required for the execution of the code sequence.

(d) Suppose the clock rate of the MIPS processor is 1 GHz. Find the CPU time of the code sequence (in terms of ns).

Answer

(a) Data hazards can be solved by forwarding: (SUB, first ADD) for R1, (first ADD, second ADD) for R5, (second ADD, SW) for R9

(b) Data hazards cannot be solved by forwarding: (LW, second ADD) for R8

(c) Total number of clock cycles = (5 – 1) + 5 + 1 = 10

(d) CPU time = 10 / 1G = 10 ns

2. Consider the following code sequence.

```
ADD  R1, R2, R3        ; R1 ← R2 + R3
ADD  R3, R4, R5        ; R3 ← R4 + R5
LW   R6, 32(R8)        ; R6 ← MEM[32 + R8]
LW   R7, 36(R8)        ; R7 ← MEM[36 + R8]
ADD  R9, R6, R1        ; R9 ← R6 + R1
ADD  R10, R7, R3       ; R10 ← R7 + R3
ADD  R11, R9, R10      ; R11 ← R9 + R10
LW   R12, 40(R8)       ; R12 ← MEM[40 + R8]
```

Suppose the code sequence is executed in a five-stage (IF, ID, EX, MEM and WB) MIPS pipeline processor with separate instruction and data memories.

(a) Determine the number of accesses to the instruction memory.

(b) Determine the number of accesses to the data memory.

(c) Suppose the execution of the code sequence produces 3 misses: 2 misses in the instruction memory, and 1 miss in the data memory. Determine the miss rate of the instruction memory.

Determine the miss rate of the data memory.

(d) Suppose the hit time of the instruction and data memories is 1 clock cycle. The miss penalty is 100 clock cycles. Based on the results in part (c), please compute the average memory access time (in terms of clock cycles) of instruction and data memories, respectively.

Answer

(a) The number of instruction memory access = 8

(b) The number of data memory access = 3

(c) Miss rate of the instruction memory = 2 / 8 = 0.25

Miss rate of the data memory = 1 / 3 = 0.33

(d) AMAT for instruction memory = 1 + 0.25 × 100 = 26 clocks

AMAT for data memory = 1 + 0.33 × 100 = 34 clocks

3. Briefly explain the following terms.

(a) Control hazard,

(b) Direct mapped cache,

(c) Fully associative cache,

(d) Translation lookaside buffer (TLB),

(e) Direct memory access (DMA).

Answer

(a) **Control hazard:** when the proper instruction cannot execute in the proper pipeline clock cycle because the instruction that was fetched is not the one that is needed.

(b) **Direct mapped cache:** A cache structure in which each memory location is mapped to exactly one location in the cache.

(c) **Fully associative cache:** A cache structure in which a block can be placed in any location in the cache.

(d) **Translation lookaside buffer (TLB):** A cache that keeps track of recently used address mappings to try to avoid an access to the page table.

(e) **Direct memory access (DMA):** A mechanism that provides a device controller with the ability to transfer data directly to or from the memory without involving the processor.

103 彰師大電子

1. (a) Describe Amdahl's Law and explain the meaning of Amdahl's Law.

   (b) Suppose you are trying to improve the performance of processor X, which spends 30% of its CPU time executing floating point (FP) operations and 16% of its CPU time executing load/store operations. The first improvement method is to make the FP operations run two times faster, and the second improvement method is to make the load/store run four times faster. Which method (the first or the second) can get higher speed-up?

   (c) Suppose the floating point (FP) operations can be improved by a factor of infinity, what is the speed-up?

Answer

(a) Amdahl's Law:

$$\text{Execution time after improvement} = \frac{\text{Execution time affected by improvement}}{\text{Amount of improvement}} + \text{Execution time unaffected}$$

Amdahl's Law stating that the performance enhancement possible with a given improvement is limited by the amount that the improved feature is used.

(b) Speedup$_1$ = $\frac{1}{\frac{0.3}{2}+0.7}$ = 1.176; Speedup$_2$ = $\frac{1}{\frac{0.16}{4}+0.84}$ = 1.136

→ The first method can get higher speed-up

(c) Speedup = $\frac{1}{\frac{0.3}{\infty}+0.7}$ = 1.428

2. Translate the following C code segment into MIPS assembly code

```
for (i = 0; i < 10; i++)
    {if (A[i] >= B[i])
        C[i] = A[i] - B[i];
     else
        C[i] = A[i] + B[i];
    }
```

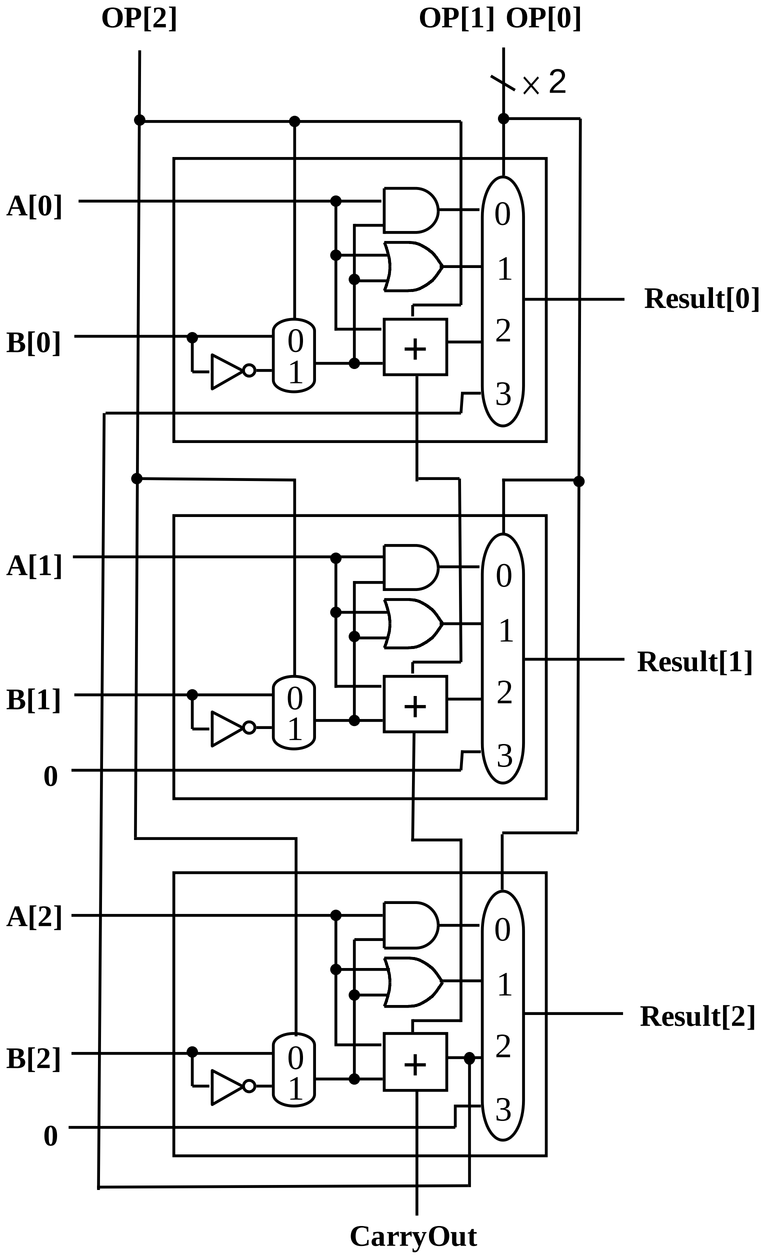
Assume variable *i* is already assigned to register ***R1***, the start address of integer array ***A*** is already stored in register ***R11***, the start address of integer array **B** is already stored in register ***R12***, the start address of integer array ***C*** is already stored in register ***R13***, and assume an integer occupies 4-byte memory.

Answer

```
            add     R1, $0, $0          ; i = 0
Loop:       slti    R2, R1, 10
            beq     R2, $0, Exit_for
            sll     R2, R1, 2
            add     R3, R2, R11         ; R3 contains address of A[i]
            add     R4, R2, R12         ; R4 contains address of B[i]
            add     R5, R2, R13         ; R5 contains address of C[i]
            lw      R6, 0(R3)           ; R6 = A[i]
            lw      R7, 0(R4)           ; R7 = B[i]
            slt     R2, R6, R7
            bne     R2, $0, Esle
            sub     R2, R6, R7          ; R2 ←A[i] – B[i]
            sw      R2, 0(R5)           ; R2 → C[i]
            j       Exit_if
Else:       add     R2, R6, R7          ; R2 ←A[i] + B[i]
            sw      R2, 0(R5)           ; R2 → C[i]
Exit_if:    addi    R1, R1, 1           ; i++
            j       Loop
Exit_for:
```

3. Design a 3-bit ALU with the following specification. The ALU has two 3-bit inputs (***A*[2:0]** and ***B*[2:0]**), one 3-bit output (***Result*[2:0]**), and a 3-bit control (***OP*[2:0]**). When *OP*==000, *Result*=*A* OR *B*; when *OP*==001, *Result*=*A* AND *B*; when OP==010, *Result*=*A* + *B*; when OP==110, *Result*=*A* - *B*; when *OP*==111, *Result*=(*A* < *B*). Design this ALU using AND gates, OR gates, NOT gates, XOR gates, MUXs (multiplexer) and FAs (full adder).

Answer



4. Write a MIPS assembly code showing how to make a function call and how to make a function return.

Answer

Make a function call: (1) Put parameters in argument registers (2) use jal instruction to jump to the starting address of the callee and simultaneously saves the address of the following instruction in the register $ra

Make a function return: (1) Put the result values in return value registers (2) use jr instruction return back to the point of origin

5. What is the difference between direct-mapped cache, two-way set-associative cache, and fully associative cache?

Answer

Direct-mapped cache is a cache structure in which each memory location is mapped to exactly one location in the cache.

Two-way set-associative cache is a cache that has two locations where each block can be placed.

Fully associative cache is a cache structure in which a block can be placed in any location in the cache.

6. Determine if each of the following statements is true (T) or false (F).

(a) In the single-cycle design, the clock cycle time for all the instructions is determined by the delay of the fastest instruction.

(b) In the multi-cycle design, the execution time for each instruction is determined by the number of clock cycles it takes.

(c) For a Moore machine, the outputs depend on both the state bits and the inputs.

(d) The ideal speedup due to pipelining is equal to the number of pipeline stages.

(e) Pipelining improves performance by decreasing the execution time of each instruction.

(f) The pipeline rate is determined by the slowest stage in the pipeline.

(g) Data hazards arise from the dependence of one instruction on an earlier one that is still in the pipeline.

(h) A method for resolving data hazards is to retrieve the missing item early from the internal resources, called forwarding.

(i) The load-use data hazard can be resolved simply by forwarding.

(j) Control hazards occur when the hardware cannot support the combination of instructions that we want to execute in the same clock cycle.

(k) In the static branch prediction, the branch instruction can be assumed to be always taken or not taken.

(1) With more hardware, the branch behavior can also be predicted dynamically during program execution.

(m) Temporal locality means that if an item is referenced, items whose addresses are close to it will tend to be referenced soon.

(n) The translation-lookaside buffer (TLB) is a memory that keeps track of recently used address translations.

(o) For the least recently used (LRU) replacement scheme, a page that has not been used for a long time is less likely to be needed than a more recently accessed page.

(p) When designing a virtual memory system, pages should be large enough to try to amortize the high access time.

(q) Having a larger number of physical pages than virtual pages is the basis for the illusion of an essentially unbounded amount of virtual memory.

(r) The page table resides in the cache and memory at the same time.

(s) Page fault occurs when an accessed page is not present in the lowest-level hard disk.

(t) The dirty bit is a bit in the page table to indicate whether a page needs to be copied back when we choose to replace it.

Answer

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
F	T	F	T	F	T	T	T	F	F
(k)	(l)	(m)	(n)	(o)	(p)	(q)	(r)	(s)	(t)
T	T	F	F	T	T	F	F	F	T

註(a)： Clock cycle time is determined by the delay of the slowest instruction

註(c)： The outputs only depend on the state bits

註(e)： Pipelining improves performance by increasing the instruction throughput

註(j)： Structure hazard

註(m)：Spatial locality

註(n)： TLB is a cache

註(q)： Having a larger number of virtual pages than physical pages is the basis for the illusion of an essentially unbounded amount of virtual memory.

註(r)： The page table resides in the memory

註(s)： Page fault occurs when an accessed page is not present in the memory

103 北科大電子

1. Please explain the following terms
   (1) Response Time
   (2) Throughput
   (3) Data Hazard
   (4) Control Hazard
   (5) Delayed branch technique
   (6) Drawing a diagram and giving a detailed explanation of 2-bit branch prediction technique

Answer

(1) The total time required for the computer to complete a task.

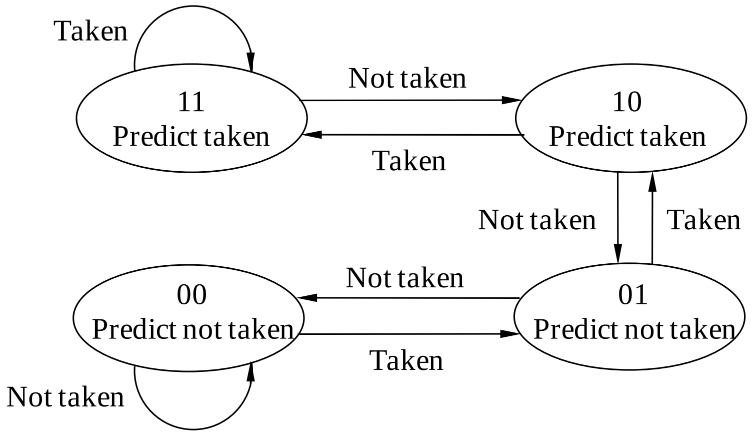
(2) The total amount of tasks done for the computer in a given time.

(3) When a planned instruction cannot execute in the proper clock cycle because data that is needed to execute the instruction is not yet available.

(4) When the proper instruction cannot execute in the proper pipeline clock cycle because the instruction that was fetched is not the one that is needed.

(5) A type of branch where the instruction immediately following the branch is always executed independent of whether the branch condition is true or false.

(6) In a 2-bit scheme, a prediction must be wrong twice before it is changed.



2. The IEEE 754 standard deals with the representation of floating point numbers in computers.
   (1) Please explain why the biased notation is used in IEEE 754 standard.
   (2) The bit pattern (1100 0111 1111 1001 0011 0000 0000 0000) is an IEEE 754 standard single precision floating point number. Please write down the related decimal number.

Answer

(1) 浮點數運算需要先比較指數大小來決定要對齊哪一個數的指數，使用bias notation可以直接比較指數欄位的無號數值即可判斷兩個浮點數指數的大小而不須考慮其正負號，因此可以加快比較速度。

(2) $(-1)^1 \times 1.11110010011 \times 2^{143-127} = - 1.11110010011 \times 2^{16} = -11111001001100000 = 127584_{10}$

3. The performance of a 1GHz processor P is measured by executing 100,000,000 instructions of benchmark code, which is found to take 0.25s. Find the MIPS and CPI for the processor P for this performance experiment.

Answer

MIPS = $\frac{100000000}{0.25\times10^6}$ = 400

CPI = $\frac{1\text{G}\times0.25}{100000000}$ = 2.5

4. Please translate the following sentences into English or Chinese
   (1) 一種常用的方法去動態預測分支(branch)是記錄每次分支(branch)是否發生或不發生，然後可以使用最近過去的行為來預測未來。
   (2) 如我們等下會看到的，大量類型的歷史紀錄，可以讓動態預測分支(branches)預測分支是否發生的正確率達90%。
   (3) When the guess is wrong, the pipeline control must ensure that the instructions following the wrongly guessed branch have no effect and must restart the pipeline from the proper branch address.
   (4) In our laundry analogy, we must stop taking new loads so that we can restart the load that we incorrectly predicted.

Answer

(1) One popular approach to dynamic prediction of branches is keeping a history for each branch as taken or untaken, and then using the recent past behavior to predict the future.

(2) As we will see later, the amount and type of history kept have become extensive, with the result being that dynamic branch predictors can correctly predict branches with more than 90% accuracy.

(3) 當猜錯時，管線控制必須確定跟在猜錯分支後的指令不會影響正確性，同時必須讓管線由適當的分支後位址重新開始。

(4) 在我們的洗衣比擬中，我們必須停止擷取新的負載(衣物)，以便重新啟動當初猜錯的負載(衣物)。