COMPUTER ARCHITECTURE EXAM QUESTIONS

QUESTION 1:

A CPU has an instruction pipeline with the following 4 segments:

1. FI (*Fetch Instruction)*
2. DA (*Decode, Address*)
3. FO (*Fetch Operand*)
4. EX (*Execution*)

The clock cycle of the pipeline is 10 ns.

Without the pipeline, one machine-language instruction would have been completed in 30 ns.

a) How long does it take to run a program of 100 instructions by this CPU with the pipeline without considering pipeline hazards?
Calculate the speedup provided by the pipeline for 100 instructions.

b) How much (clock cycles) is the branch penalty in this system if a misprediction occurs? Why?

c) Assume that there is a loop with 10 instructions that contains only one conditional branch instruction (1 branch + 9 other instructions). This loop runs 100 times.
The system uses the static branch prediction strategy "Always predict NOT taken".
Assume that the ratio of taken branches is 70% and for not taken branches 30%.
How long does it take to run this program (100 x loops)?
Calculate the speedup obtained for this program.

QUESTION 2:

A RISC CPU has an instruction pipeline with 3 stages: I (Fetch), A (Decode/ALU), D (Data).
The given program on the right performs a multiplication (C=AxB) and clears the registers that hold the source operands A and B.
A starts in memory at the address $500, B at $502, and C at $504.
Remember R0 of the CPU always contains zero.

```
START:
      LDSU  (R0)$500, R10    ; R10 <-- A (16-bit)
      ADD   R0, 0, R11       ; R11 <-- 0
      LDSU  (R0)$502, R11    ; R11 <-- B (16-bit)
      ADD   R0, 0, R12       ; R12 <-- 0 (C)
LOOP:
      ADD   R10, R12, R12    ; R12 <--R12+A
      SUB   R11, 1, R11      ; Decrement R11
      JMPR  BHI, LOOP        ; If higher than zero
      ADD   R0,R0,R10        ;R10 <-- 0 (Clear A)
      STL   (R0)$504,R12     ;M[504]<-- Result (C)
```

a) Examine the given program by drawing the timing diagram of the 3-stage pipeline.
Is there any data conflict?

b) Explain the control (branch) conflict encountered in the given program. What happens if this branch conflict is not solved?

c) Find a software solution to the branch problem without decreasing the performance of the pipeline. Do not change the algorithm.

INSTRUCTION SET:

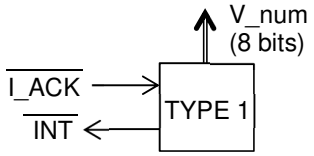
ADD	Rs,S2,Rd	*Rd* ← *Rs* + S2	
SUB	Rs,S2,Rd	*Rd* ← *Rs* - S2	
LDSU	(Rs)S2,Rd	*Rd←M[Rs + S2]*	Short unsigned
STL	(Rs)S2, Rm	*M[Rs + S2]* ← *Rm*	Store long
JMPR	COND,Y	*PC←PC + Y*	Jump relative

COMPUTER ARCHITECTURE EXAM QUESTIONS

QUESTION 3:

In a MC68000-based system there are 3 interrupting devices of two different types.

TYPE 1: To send an interrupt request it asserts the INT′ (active low) output. When its I_ACK′ input becomes zero it puts its vector number on the V_num lines. The maximum delay between I_ACK′ and V_num is 20 ns.



When the I_ACK′ input is one the V_num lines are in 3rd state (high impedance).

TYPE 2: This type of device has all lines of TYPE 1 device (INT′ , I_ACK′ , V_num). In addition it has an V_ACK′ output. When its I_ACK′ input becomes zero it puts its vector number on the V_num lines and asserts the V_ACK′ output (active low). When the I_ACK′ input becomes one the V_ACK′ output is deactivated and the V_num lines transit to the 3rd state (high impedance).

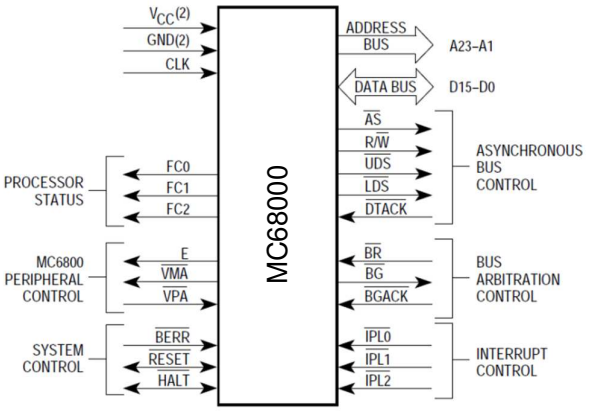
Note that the delay between I_ACK′ and V_num is not determined for this type.

An interrupting device (A) of TYPE 1 and two devices (B, C) of TYPE 2 will be connected to the 2nd interrupt level of the MC6800 using a serial interrupt priority controller (*daisy chain*). A has the highest priority and C the lowest. Assume that each logic gate has the propagation delay of 5 ns.

a) Design and draw the internal structure only of a single link of the daisy chain that can be used with the given interrupting devices.

b) Design and draw the entire system. Do not show the internal structure of a link of the daisy chain.

c) What happens if the device A sends an interrupt request while the interrupt service routine (ISR) of the device C is running.



QUESTION 4:

A CPU with 8-bit data bus, has two Interrupt Request inputs (**IRQ1** , **IRQ2**) and two related Interrupt Acknowledgement outputs (**INTA1**, **INTA2**). All signals are active at "1". If the CPU receives a request from any of the inputs (IRQx), it works with vectored interrupts and reads the interrupt vector number after the acknowledgement (INTAx=1) of the related interrupt when its Data acknowledgement input (**DACK**) is "1".

IRQ1 has higher priority than the IRQ2 (If there are simultaneous requests from both inputs, IRQ1 is accepted, INTA1=1).

In the interrupt (housekeeping) cycle of the IRQ1, interrupts from both inputs are disabled by clearing interrupt masks (IM1=0, IM2=0). Hence, interrupt service routines related to IRQ1 cannot be interrupted by other requests.

In the interrupt cycle of the IRQ2, interrupts only from IRQ2 are disabled by clearing the interrupt mask (IM2=0). Hence interrupt service routines related to IRQ2 can be interrupted only by requests from IRQ1.

In this system, there are three devices that work as interrupt sources (*IS1*, *IS2*, *IS3*).

Each of these devices has an Interrupt Request output (**IRQ**), an Interrupt Acknowledgement input (**INTA**), and 8-bit vector number output (**VN**). 20 ns after the interrupt has been acknowledged (**INTA=1**) the device outputs its vector number at the **VN** and removes its request (**IRQ=0**).

Priority (precedence) order of the devices: *IS1* > *IS2* > *IS3*

The interrupt service routine related to *IS1* cannot be interrupted by *IS2* or *IS3*. Interrupt service routines related to *IS2* and *IS3* can be interrupted only by *IS1*.

a) Design and draw the system with the CPU, 3 devices (*IS1*, *IS2*, *IS3*) and the necessary logic gates. Write the logic expressions for the interrupt request inputs of the CPU (IRQ1 , IRQ2) and for the interrupt acknowledgement inputs of the interrupt sources (INTA_IS1, INTA_IS2, INTA_IS3).

b) Instruction cycle of the CPU has the following 4 states (phases):

1. Instruction fetch and decode: 60ns, 2. Operand fetch: 70ns, 3. Execution: 50ns, 4. Interrupt: 200ns.

Durations of all interrupt service routines (ISR) are 2000ns.

COMPUTER ARCHITECTURE EXAM QUESTIONS

Assume that we start a clock (Clock = 0) when the CPU begins to run the program and all interrupts are enabled (IM1=1, IM2=1).
At Clock = 10ns the device *IS2* sends an interrupt request and then at Clock = 300ns the device *IS1* sends an interrupt request.
When (Clock =?) will the ISRs of the I*S1* and *IS2* start to run? Why?

QUESTION 5:
Instruction cycle of a CPU has the following 4 states (cycles) with the given durations.
1. Instruction fetch: 60 ns, 2. Instruction Decode: 20 ns, 3. Operand fetch: 60 ns, 3. Execution: 30 ns,
4. Interrupt (if necessary): 200 ns.
Note that the CPU accesses the memory in instruction fetch and operand fetch cycles but not in the decode and execution cycles. The CPU enters the interrupt cycle only if there are interrupt requests and housekeeping operations (saving return address, taking the vector address, etc.) take 200 ns.
Memory access time and I/O interface access times are both 50 ns.

In this system there is a 2-wire DMAC (*Direct Memory Access Controller*) that is configured to transfer words from the I/O interface to the memory using the **cycle-stealing** technique. The type of the DMAC is **fly-by** (Implicit). Data don't pass through DMAC.

The processor will run a program with 10 instructions and the DMAC is configured to transfer 10 bytes from the I/O interface to the memory. Assume that we start a clock (Time=0) when the CPU begins to run the program and as the CPU is in the instruction fetch cycle for the first instruction (Time=5ns) the DMAC attempts to start the data transfer.

d) When (Time =?) will the DMAC complete the transfer of the first byte? Why?
e) When (Time =?) will the CPU finish the first instruction? Why?
f) When (Time =?) will the DMAC complete the transfer of all 10 bytes? When (what Time) will the CPU complete the run of the entire program?
g) Assume that, instead of the DMA technique, **interrupt-driven I/O** technique is used to transfer 10 bytes from the I/O interface to the memory.
The interrupt service program transfers one word each time and it takes 500 ns (Housekeeping operations are not included).
Assume that, as the CPU is in the instruction fetch cycle for the first instruction (Time=5ns) the first interrupt request arrives from the I/O interface.
When (Time =?) can be the first word transferred from the I/O interface to the memory? Why?
h) When (Time =?) will be all the 10 bytes transferred and the main program of 10 instructions finished when the **interrupt-driven** technique is used?

QUESTION 6:
a) Note: in RAID 4, disks operate independently (not synchronized). Large strips (blocks) are used.
Draw a RAID 4 system with total 4 disks (data + parity) and distribute 6 blocks (block 0 – block 5) over the disks.
Assume that the access time of a disk is **ta**.
i) How long does it take to read words from two blocks (for example block 0 and block 4) in two different disks? (One word from block 0, one word from block 4)
ii) How long does it take to update (write) words of two blocks (for example block 0 and block 4) in two different disks? Explain. (One word in block 0 and one word in block 4)
b) Answer the question in **a)** (i and ii) for the RAID 5 system.

QUESTION 7:
A computer system includes **512 KB** (B: Byte) main memory and a cache memory that can hold **4 KB** data. Data transfers between main and cache memories are performed using blocks of **32 bytes**.
In necessary cases LRU is used as replacement technique.
Answer this question according to following two different mapping techniques:

a) Direct mapping,
b) 2-way set associative (2 blocks in a set)

i) In what fields is the physical address divided by the cache control unit? Give the lengths of the fields.
ii) What is the size of the tag memory (how many rows, the length and contents of each row)?
iii) Assume that the cache memory is empty and the following addresses are generated by the CPU in the given order. List all operations performed by the cache controller. Show the values written to the tag memory.
1- $00053, 2- $01050, 3- $0004F

QUESTION 8:
A computer system includes 1 MB (B: Byte) main memory and a cache memory that can hold 8 KB data. Data transfers between main and cache memories are performed using blocks of 32 bytes.
In necessary cases LRU is used as replacement technique.
Answer this question according to following three different mapping techniques:
c) Full associative
d) Direct mapping,
e) 4-way set associative (4 blocks in a set)

i) In what fields is the physical address divided by the cache control unit? Give the lengths of the fields.
ii) What is the size of the tag memory (how many rows, the length and contents of each row)?
iii) Assume that the cache memory is empty and the following addresses are generated by the CPU in the given order. List all operations performed by the cache controller. Show the values written to the tag memory.
2- $00025, 2- $0003F, 3- $02020

QUESTION 9:
In a symmetric multiprocessor (SMP) system with a shared bus, there are two CPUs (CPU1 and CPU2) that have local cache memories. The system does not include a shared L2 cache.
Assume that there is a shared variable A in the system. The valid copy of A resides **only** in the cache of CPU1 (A=1). The copies of A in the main memory and in the cache of CPU2 are not valid (A=2).
To provide cache coherence the snoopy **MESI** protocol is used.
a) What are the current states of cache frames (in CPU1 and CPU2) that contain the A, if the system is in this described situation?
b) Assume that the system is in the situation in a). List the operations performed by the MESI protocol when the CPU2 attempts to read the A.