Question 1: HW-3 (2018)

A **CPU** with **8-bit** data bus, has an Interrupt Request input (**IRQ**) and an Interrupt Acknowledgement output (**INTA**). Both signals are active at "1". If the vectored/autovectored input (**VA**) of the **CPU** is "**0**", the **CPU** works with vectored interrupts, so that it reads the interrupt vector number after the acknowledgement of the interrupt, when it's Data acknowledgement input (**DACK**) is "**1**". The **CPU** also supports autovectored interrupts. After the acknowledgement of an interrupt request (**INTA**=**1**) if the vectored/autovectored input (**VA**) of the **CPU** is made "**1**", then it does not read a vector number and works in autovectored mode.

In this system there four interrupt sources (**A1, A2, B1, B2**) of two different types (**A, B**).

• **Type A:** These devices (**A1, A2**) have an Interrupt Request output (**IRQ**) and an Interrupt Acknowledgement input (**INTA**). They don't have a vector number outputs. They work in autovectored mode.
• **Type B:** These devices (**B1, B2**) have an Interrupt Request output (**IRQ**), an Interrupt Acknowledgement input (**INTA**), and 8-bit vector number output (**VN**).

Priority (precedence) order of the devices: **A1 > A2 > B1 > B2** (Read carefully!)

a) Design and draw the system with the **CPU**, **4 devices** (**A1, A2, B1, B2**) and the priority interrupt controller. First, show the priority interrupt controller only **as a box**. Then design and draw the internal structure of the priority interrupt controller using logical gates.

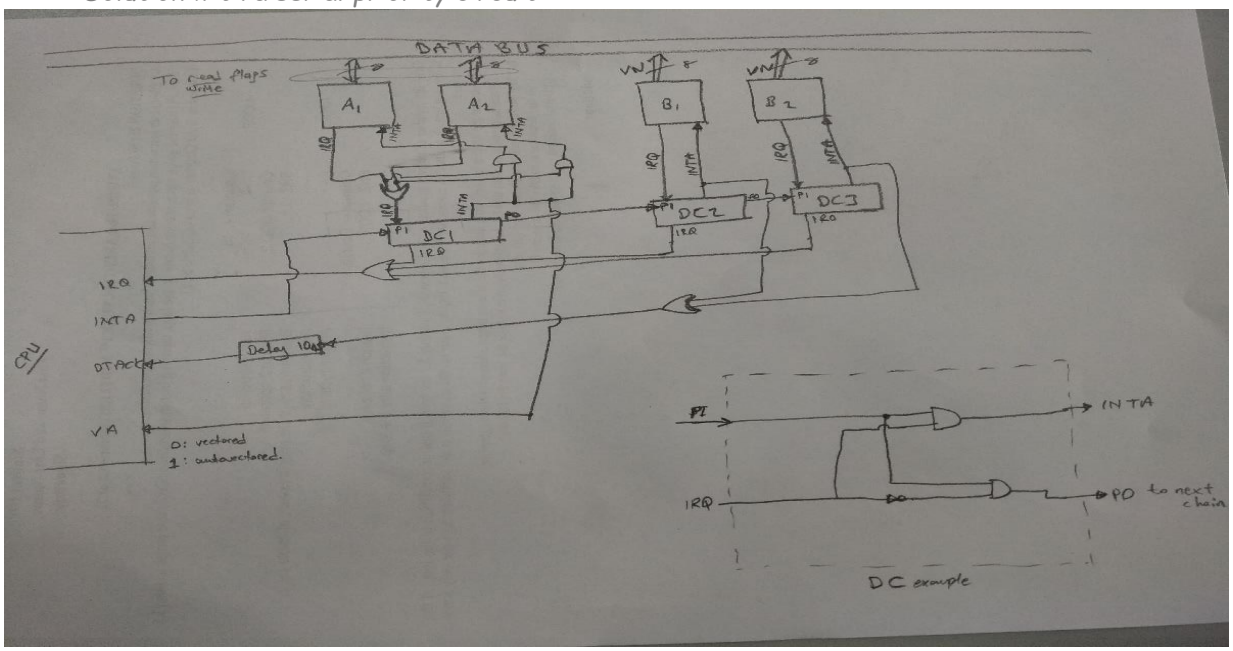
b) Assume that the devices **A1** and **B1** assert their interrupt requests at the same time. Show step by step all the signals that are sent in the system until the requests of both devices has been fulfilled.

c) How does the **CPU** determine the start address of the interrupt service routine to be run if the interrupt source is a device of type **A** or of type **B** ?

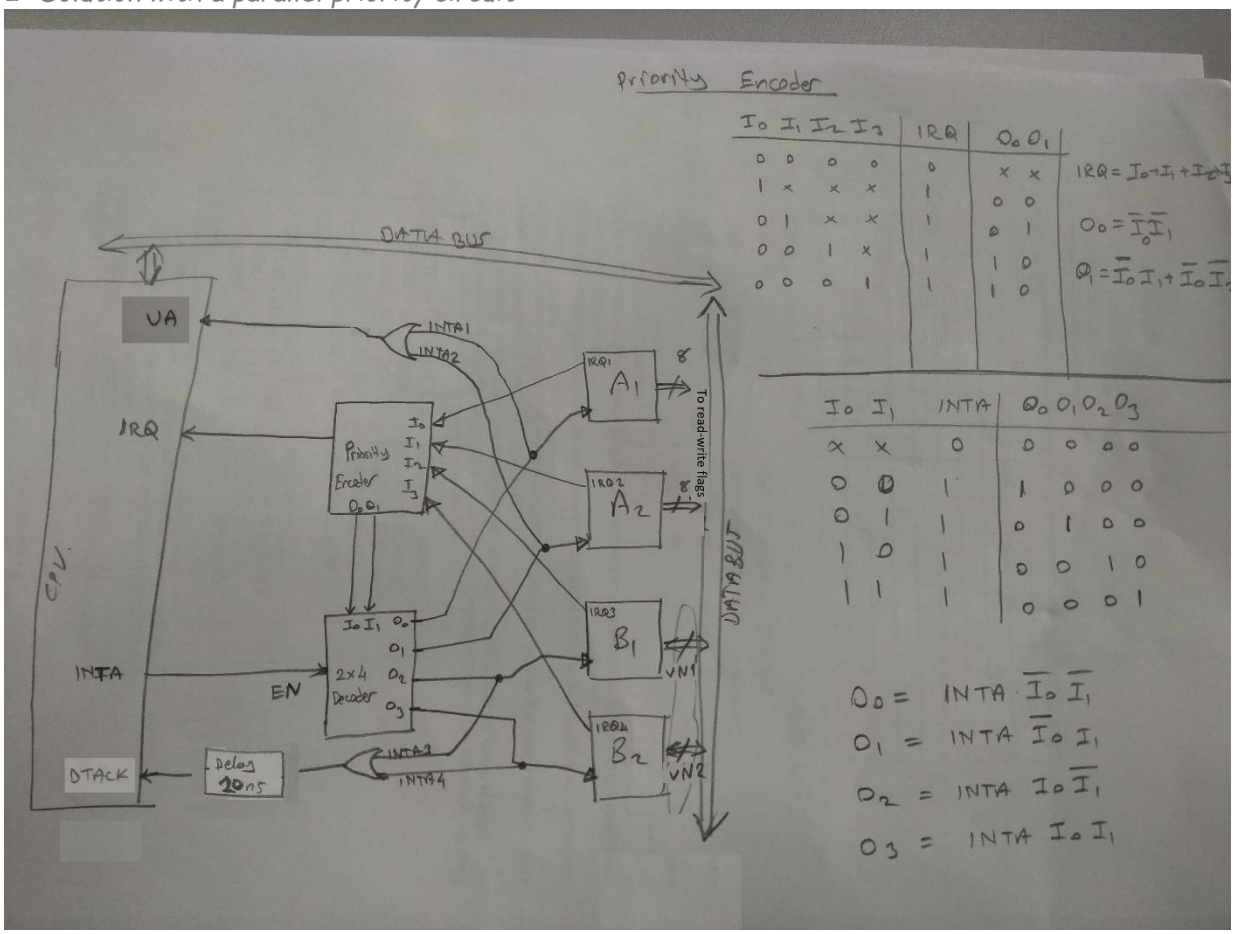
Solution 1:

a) (50 Points)

1- Solution with a serial priority circuit



2- Solution with a parallel priority circuit



b) (30 Points)

IRQ -> CPU
CPU-> INTA
INTA1 -> A1 (IRQ1 disabled)
INTA -> VA

CPU makes interrupts disabled (DI).
ISR tests interrupt from A1 by autovectored software polling
CPU executes ISR of A1
CPU makes interrupts enable (EI)

Interior operations

---Execute one instruction from interrupted program---

IRQ -> CPU
CPU-> INTA
INTA3 -> B1 (IRQ3 disabled)
INTA -> DTACK (20 ns delayed)
VN -> DATABUS (20 ns delayed)

CPU makes interrupts disabled (DI).
CPU executes ISR of B1
CPU makes interrupts enable (EI)

Interior operations

---Return to interrupted program----

c) (20 Points)

To get **ISR** addresses of **A1** and **A2**, **CPU** jumps to the autovector **ISR**. This common **ISR** checks flags of **A1**. If not asserted checks flags of **A2**. Then jumps to related **ISR** (driver program).

CPU access the **ISR** addresses of **B1** and **B2** from vector table using **VN1** and **VN2**.

Question 2:

Instruction cycle of a CPU has the following **4 states** (cycles):
1. Instruction fetch, 2. Operand fetch, 3. Execution, 4. Interrupt
In this system there is a **2-wire DMAC** (Direct Memory Access Controller) that is configured to transfer **10 words** from the **I/O interface** to the memory using the **cycle-stealing** technique.

Assume that the **CPU** is running a main program that reads operands from the memory but do not need to access the memory in the execution cycles.
a) Assume that, as the **CPU** is in the instruction fetch cycle for the first instruction the **DMAC** attempts to start the data transfer.

When can the **DMAC** transfer the first word from the **I/O interface** to the **memory**? Give your answer according to the machine language instructions and the cycles executed by the **CPU**. (Before fetch, after fetch, in the execution cycle, after the execution of the first instruction etc.) Why?
b) How many instructions of the main program can be executed by the **CPU** during the transfer of **10 words**? Explain.
c) Assume that, instead of the **DMA** technique, **interrupt-driven I/O technique** is used to transfer **10 words** from the **I/O interface** to the **memory**.

The interrupt service program transfers one word each time it runs.
Assume that, as the **CPU** is in the **instruction fetch cycle** for the first instruction the first interrupt request arrives from the I/O interface.
When can the first word transferred from the **I/O interface** to the **memory**? Give your answer according to the machine language instructions and the cycles executed by the **CPU**. (Before fetch, after fetch, in the execution cycle, after the execution of the first instruction etc.) Why?

d) How many instructions of the main program can be executed by the **CPU** during the transfer of **10 words** using the interrupt-driven technique? Explain.

Solution 2:

a) The **CPU** first completes the current bus cycle then grants the bus to the **DMAC**.
Therefore the **CPU** first completes the instruction fetch cycle. The first word is transferred after the instruction fetch cycle (before the operand fetch cycle).
b) Because of the cycle-stealing technique the **DMAC** transfers one word and gives the bus back to the CPU.
CPU: **DMAC**:
IF -
Transfer
OF - (Because of cycle-stealing; in burst mode the DMAC would continue to transfer)
EX Transfers (simultaneously)
Two words can be transferred in one instruction cycle. Therefore **5 instructions** of the main program can be executed by the **CPU** during the transfer of **10 words**.
Note: The **CPU** does not enter the interrupt cycle because there is not any interrupt request in **b**).
Remember; the **CPU** checks the interrupt request lines at the end of the execution cycle. If there is a request and if it is not masked the **CPU** enters the interrupt cycle. Otherwise the **CPU** continues with the instruction fetch cycle of the next instruction.
c) The **CPU** checks the interrupt request lines at the end of the execution cycle. If there is a request to be accepted, the starting address of the interrupt service routine (**ISR**) is determined in the interrupt cycle (vectortable). Then the **CPU** goes to the fetch cycle and fetches first instruction of the **ISR**. The first word is transferred by the **CPU** in the execution cycle of the transfer instruction in the **ISR**.

The following answer is also accepted: The first word is transferred after the execution of the currents instruction.

d) After each instruction of the main program only one word is transferred. Therefore **10 instructions** of the main program can be executed by the **CPU** during the transfer of **10 words**.