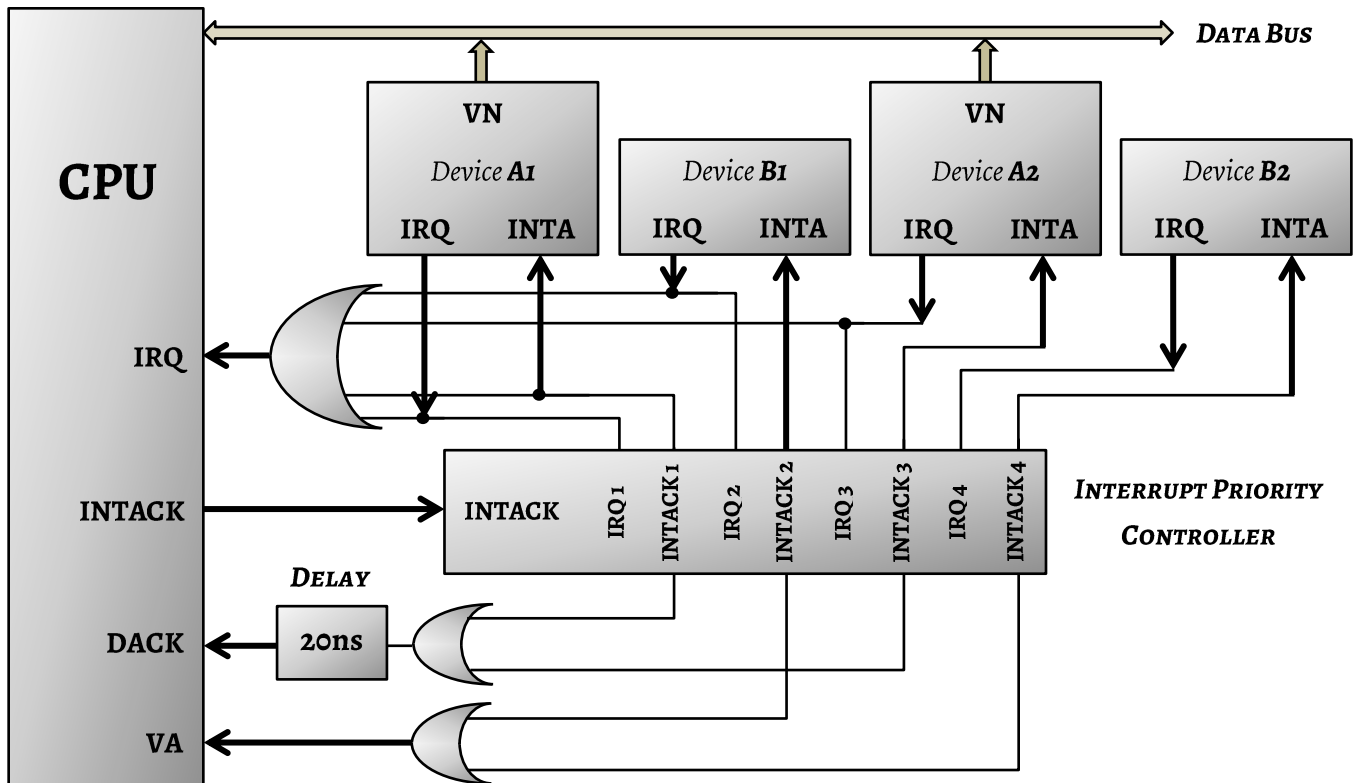
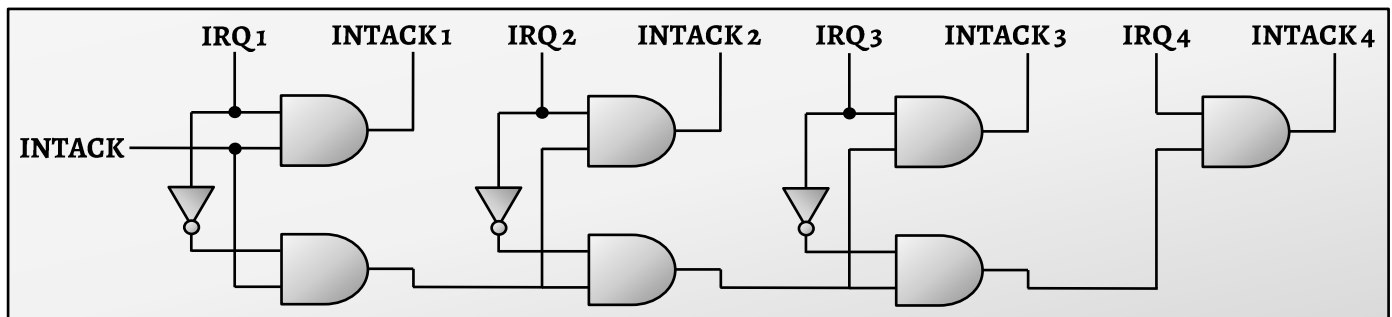


COMPUTER ARCHITECTURE 2ND HOMEWORK — SOLUTION

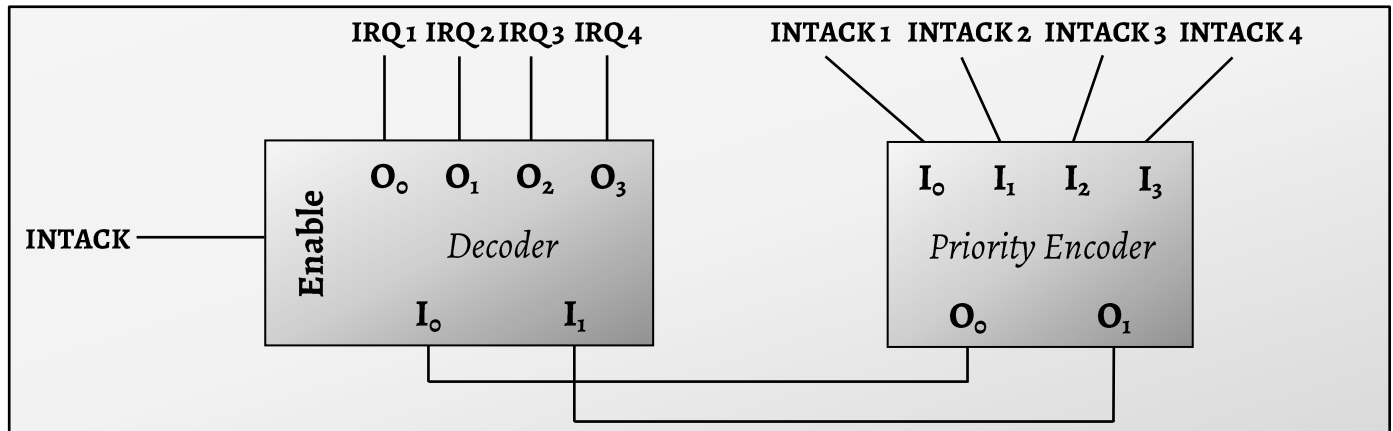
1.a) CPU/Device Connections



1.b) Serial Interrupt Priority Controller



1.c) Parallel Interrupt Priority Controller



2x4 DECODER

I_0	I_1	Enable	O_0	O_1	O_2	O_3
x	x	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

$$O_0 = \text{Enable} \cdot I_0' \cdot I_1'$$

$$O_1 = \text{Enable} \cdot I_0' \cdot I_1$$

$$O_2 = \text{Enable} \cdot I_0 \cdot I_1'$$

$$O_3 = \text{Enable} \cdot I_0 \cdot I_1$$

4x2 PRIORITY ENCODER

I_0	I_1	I_2	I_3	O_0	O_1
0	0	0	0	x	x
1	x	x	x	0	0
0	1	x	x	0	1
0	0	1	x	1	0
0	0	0	1	1	1

$$O_0 = I_0' \cdot I_1'$$

$$O_1 = I_0' \cdot (I_1 + I_2')$$

2)

Determining the Start Address of the ISR

Type A: As the interrupt is acknowledged, the device places the start address of the driver program on the data bus. After a delay of 20ns, the DACK signal is activated and the CPU reads the address from the bus.

Type B: As the interrupt is acknowledged, the VA signal is activated and the CPU branches to the autovectorized ISR. Within that ISR, the software polling flags of the devices are checked in priority order (first B₁, then B₂), and then the CPU jumps to the driver program associated with the first encountered device with an active request.

KHP'nin Başlangıç Adresini Saptama

A Tipi: Kesme kabul edildiğinde cihaz sürücü programının başlangıç adresini veri yoluna koyar. 20 ns'lik bir gecikmeden sonra DACK sinyali etkinleştirilir ve MİB adresi yoldan okur.

B Tipi: Kesme kabul edildiğinde VA sinyali etkinleştirilir ve MİB otovektörlü KHP'ye dallanır. Bu KHP'nin içinde cihazların yazılımsal yoklama bayrakları öncelik sırasına göre kontrol edilir (önce B₁, daha sonra B₂) ve MİB etkin bir isteğiyle karşılaşılana ilk cihazla ilişkilendirilmiş sürücü programına gider.
