



Student #	Name	Signature

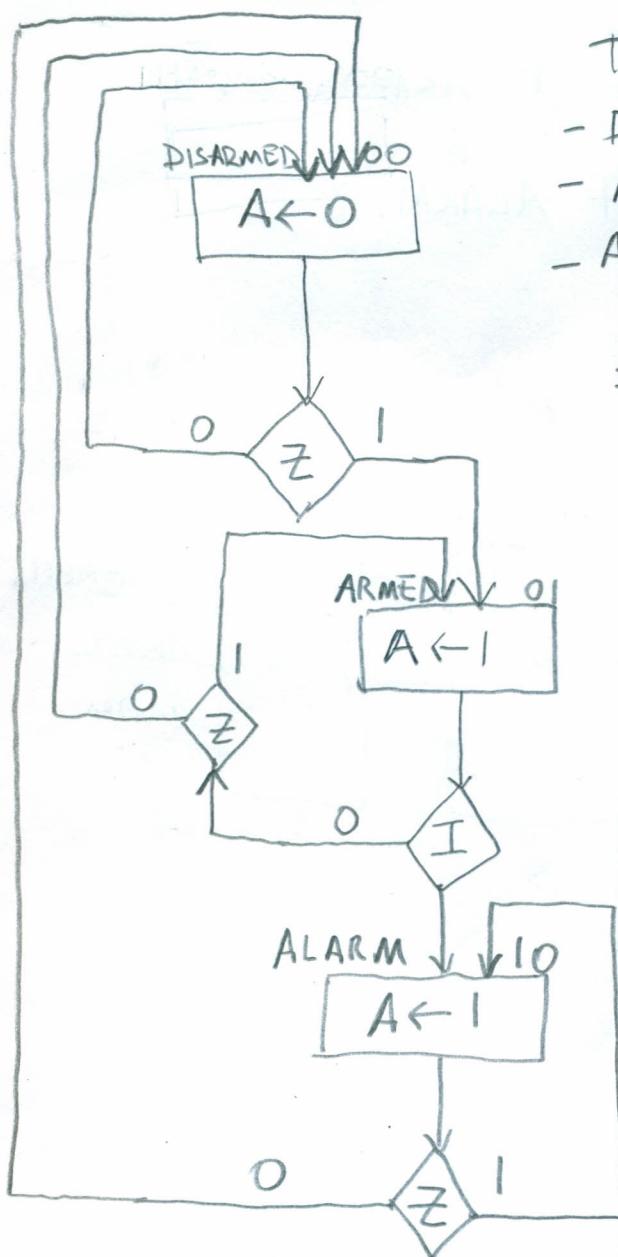
Q1:[30 pts] Design a security system for a building with the following specifications:

- The building has an alarm system activated with an input signal Z.
 - The Z input activates the sensors of the building with an output signal A, and sets the alarm system.
 - The alarm system can be de-activated anytime by the authorized personnel.
 - If an intrusion is detected with digital input signal I, the alarm starts and continues until the authorized personnel disarm the system.
- a) [15 pts] Draw the Algorithmic State Machine (ASM) chart for this security system.
b) [15 pts] Implement the control unit of the security system with the least amount of D flip-flops.
You may use decoder to simplify the circuit.

a)

NOTE :
THIS IS
ONLY ONE
SOLUTION

THERE WERE
OTHER
SOLUTIONS
THAT
RECEIVED
FULL
CREDIT



There are 3 states
- DISARMED
- ARMED
- ALARM

Z: Input arms, dearms,
and de-activates.

I: Input detects
intruder

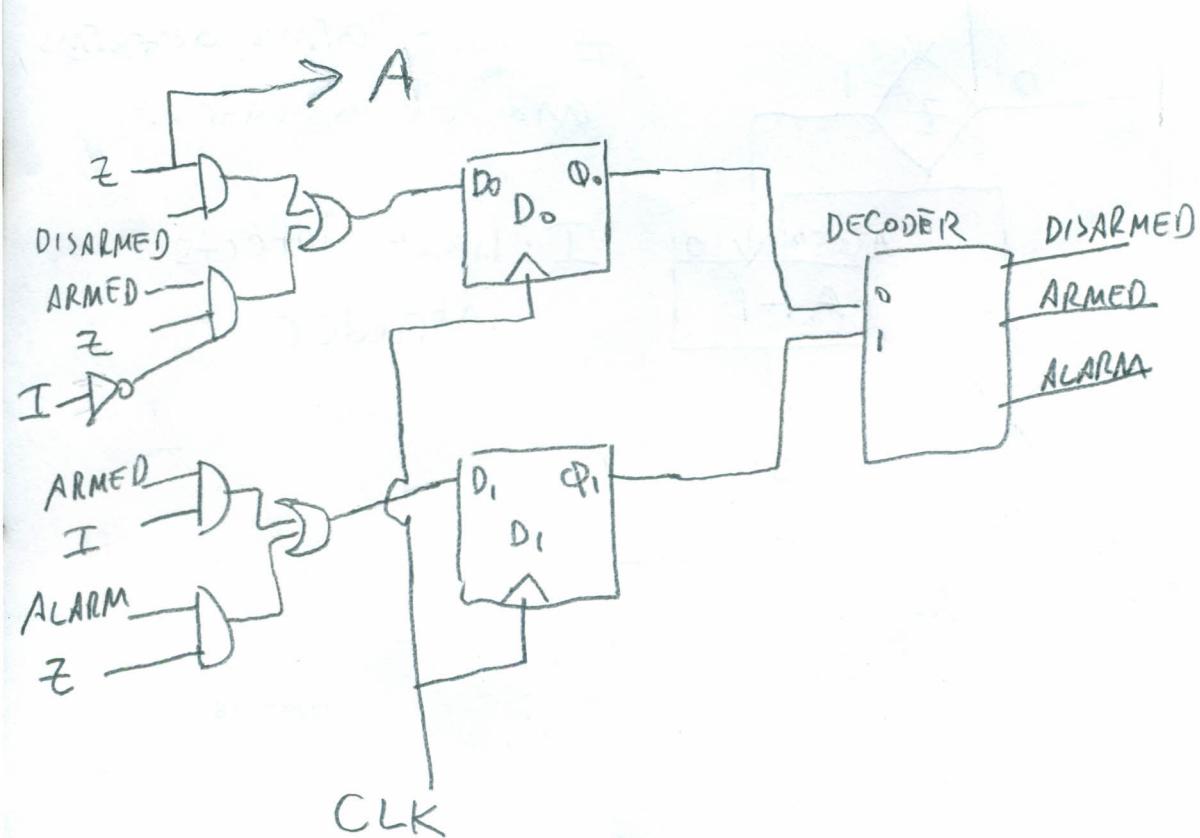
b)

PRESENT STATE	INPUT	INPUT		NEXT STATE		OUTPUT
		Q1	Q0	Z	I	
DISARMED		0	0	0	X	DISARMED
		0	0	1	X	ARMED
ARMED		0	1	0	0	DISARMED
		0	1	X	1	ALARM
		0	1	1	0	ARMED
ALARM		1	0	0	X	DISARMED
		1	0	1	X	ALARM

Do : DISARMED.Z + ARMED.Z.I

D1 : ARMED.I + ALARM.Z

A : A = Z





Student #	Name	Signature

Q2:[30 pts] The Figure Q2-1 shows the microprogram controlled CPU configuration, the instruction format, and the microinstruction format for the CPU. The F1, F2, and F3 fields specify the microoperations, the CD field specifies the condition, and the BR field specifies the branch controls as defined in Table Q2-1. The Opcode of the instruction is mapped to the control memory with Oxxxx0 template. The FETCH and INDIRECT microcodes exist in the control memory starting from address \$40.

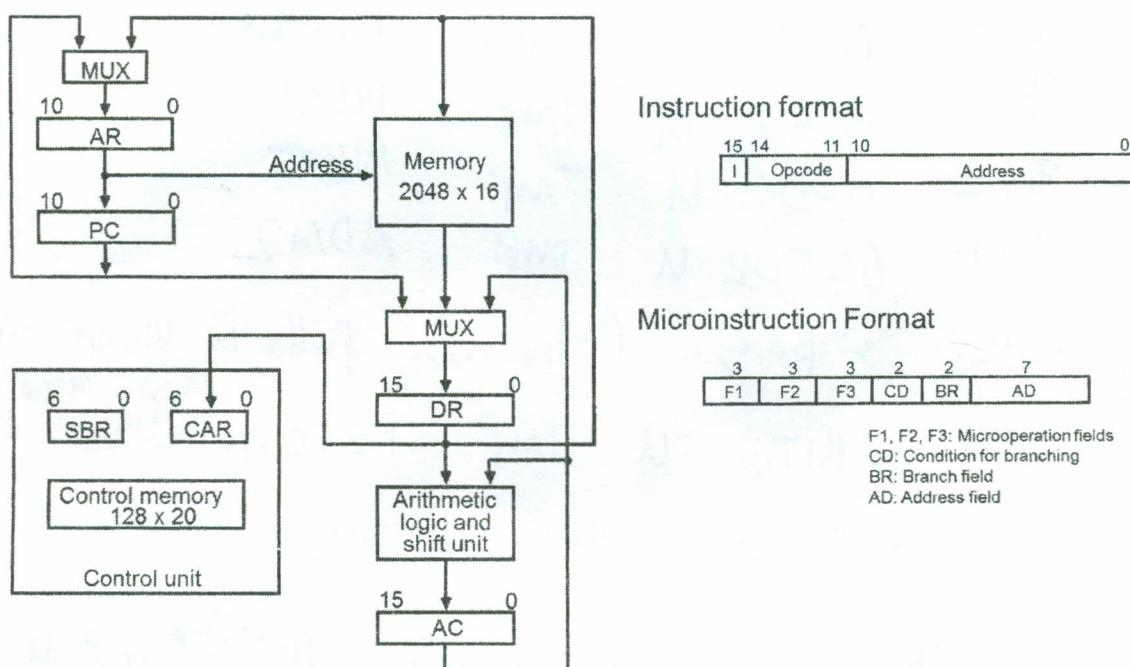


Figure Q2-1: Microprogram controlled CPU configuration, instruction and microinstruction format

- a) [15pts] Write the symbolic microprogram for the following instructions. Indicate the address in the control memory for each instruction. Assume the FETCH and INDIRECT microcodes are present at addresses *FETCH* and *INDIRECT*, and need not be re-written.

OPCODE 0100 - AND: AC <- AC AND M[EA]
OPCODE 0101 - CMP (Complement): AC <- AC'
OPCODE 0111 - ADM (Add to memory) : M[EA] <- M[EA]+AC

- b) [15pts] Write the binary microcode for the instructions in Part a. Indicate the address in the control memory for each instruction.

NOTE: You can use table: "Table Q2-1: Symbols and Binary Code for Microinstruction Fields" on Page 8.

a) ADR. ORG \$10 = 16 = 10H

AND \$10 NOP I CALL INDIRECT
\$11 READ U JMP NEXT
\$12 AND U JMP FETCH

b) Convert symbolic code using the provided table.

CMP ADR. ORG \$14
\$14 COM U JMP NEXT
\$15 Nop U JMP FETCH

ADM ADR. ORG \$1C
\$1C NOP I CALL INDIRECT
\$1D READ U JMP NEXT
\$1E ADD U JMP NEXT
\$1F ACTDR U JMP ADM2

ADM2 ORG > \$45 (otherwise falls in OPCODE Memory Map area)
WRITE U JMP FETCH

Notes:

- ① Filling the unused area with NOP NOP NOP U JMP FETCH is a good design practice, if CAR mistakenly falls into these addresses the machine goes to the next instruction.
- ② For ADM2 one can use the unused lines after CMP for a more economical design.
- ③ Some people did DRTAC & ACTDR and ACTDR & DRTAC to preserve the value of AC.
This was not necessary. IN THEORY THIS IS POSSIBLE, IN PRACTICE DON'T DO THIS! IN REAL LIFE THERE WILL BE NON-ZERO GATE DELAYS FOR MUX AND ESPECIALLY ALU. THIS MAY CORRUPT DATA TRANSFER. RELAX WITH ONE MORE CLK CYCLE



İ.T.Ü. Faculty of Computer and Informatics
BLG222E Computer Organization, Final
Asst. Prof. Dr. Erdem Matoglu, Asst. Prof. Dr. Berk Canberk, Res. Asst. Selda Uyanık

05.30.2012
Duration: 120 mins

Student #	Name	Signature

Q4 : a) [5pts] Explain the difference between a hardwired and microprogrammed control CPU.

THE ANSWERS OF THIS PART
IN LECTURE SLIDES

b) [5pts] Explain the difference between the Von-Neumann and Harvard architecture.

SEE LECTURE SLIDES

c) [BONUS 5pts] Explain briefly the difference between a RISC and CISC machine.

SEE LECTURE SLIDES

Soru3:[30 Puan] 8 bitlik 4 adet saklayıcı (A,B,C ve D) bir ortak veri yolu üzerinden birbirlerine veri aktarmaktadırlar. A,B ve C saklayıcıları, aralarında verilerini değiştirmek için (swapping) ortak bir veri yolu üzerinden D saklayıcısını kullanmaktadır. A,B ve C saklayıcıları, aşağıdaki 3 değişik komutu kullanarak birbirleriyle verilerini değiştirmektedirler:

-Komut 0: A ile B arasında veri değişimi yapılmaktadır.

-Komut 1: A ile C arasında veri değişimi yapılmaktadır.

-Komut 2: B ile C arasında veri değişimi yapılmaktadır.

İlgili komutlar lojik bağlantınlarda S₁,S₀ ikili değişkenleriyle temsil edilecektir. Bu ortak veri yolu sistemini sürmek için, gerekli sayıda üç-durumlu tampon devreler kullanılacak ve bu tamponların sürme kontrol girişleri DR_X değişkeniyle gösterilecektir. Saat periyotlarına ilişkin değerler T₁,T₂,.. ile gösterilecektir.

- a) Her bir saat darbesinde saklayıcılarda meydana gelen değişiklikleri yazınız.
- b) Saklayıcıların yükleme girişlerinin (LD_A , LD_B , LD_C , ve LD_D) ve üç-durumlu tamponların sürme kontrol girişlerinin (DR_X) lojik bağlantılarını bulunuz.
- c) Bu üç komutu gerçekleştiren ortak veri yolu sistemini a ve b adımlarında elde ettiğiniz bilgileri de kullanarak tasarlaymentınız.

⑤ Cevap: Cevap:

⑦

Cevap:

	S, So	$S \rightarrow A \wedge S' \rightarrow B$
Konut 0	0 0	$\Rightarrow S, S'_0$
Konut 1	0 1	$\Rightarrow S, S'_0$
Konut 2	1 0	$\Rightarrow S, S'_0$

Konut 0: $A \leftrightarrow B$

$$S, S'_0 \cdot T_1: D \leftarrow A$$

$$S, S'_0 \cdot T_2: A \leftarrow B$$

$$S, S'_0 \cdot T_3: B \leftarrow D$$

Konut 1: $A \leftrightarrow C$

$$S, S'_0 \cdot T_1: D \leftarrow A$$

$$S, S'_0 \cdot T_2: A \leftarrow C$$

$$S, S'_0 \cdot T_3: C \leftarrow D$$

Konut 2: $B \leftrightarrow C$

$$S, S'_0 \cdot T_1: D \leftarrow B$$

$$S, S'_0 \cdot T_2: B \leftarrow C$$

$$S, S'_0 \cdot T_3: C \leftarrow D$$

* A'ya veri yüklemek için: (LD-A'nın losjh ifadesi) ②

$$\left. \begin{array}{l} S_1'S_0'T_2 : A \leftarrow B \\ S_1S_0.T_2 : A \leftarrow C \end{array} \right\} \boxed{LD-A = S_1'S_0'T_2 + S_1S_0^T T_2}$$

* B'ye veri yüklemek için: (LD-B'nin losjh ifadesi)

$$\left. \begin{array}{l} S_1'S_0^T.T_3 : B \leftarrow D \\ S_1S_0^T.T_2 : B \leftarrow C \end{array} \right\} \boxed{LD-B = S_1'S_0^T.T_3 + S_1S_0^T T_2}$$

* C'ye veri yüklemek için: (LD-C'nin losjh ifadesi)

$$\left. \begin{array}{l} S_1'S_0.T_3 : C \leftarrow D \\ S_1S_0^T.T_3 : C \leftarrow D \end{array} \right\} \boxed{LD-C = S_1'S_0T_3 + S_1S_0^T T_3}$$

* D'ye veri yüklemek için: (LD-D'nin losjh ifadesi)

$$\left. \begin{array}{l} S_1'S_0^T T_1 : D \leftarrow A \\ S_1S_0 T_1 : D \leftarrow A \\ S_1S_0^T T_2 : D \leftarrow B \end{array} \right\} \boxed{LD-D = S_1'S_0^T T_1 + S_1S_0 T_1 + S_1S_0^T T_2}$$

A \rightarrow A : ST.losjh

B \rightarrow B : ST.losjh

C \rightarrow C : ST.losjh

* A'dan veri okumak için (DR-A'nın logik ifadesi) ③

$$\left. \begin{array}{l} S_1' S_0 T_1 : D \in A \\ S_1' S_0 T_1 : D \in A \end{array} \right\} \boxed{DR_A = S_1' S_0 T_1 + S_1' S_0 T_1} \quad [5]$$

* B'den veri okumak için (DR-B'nin logik ifadesi)

$$\left. \begin{array}{l} S_1' S_0' T_2 : A \in B \\ S_1 S_0' T_1 : A \in C \end{array} \right\} \boxed{DR_B = S_1' S_0' T_2 + S_1 S_0' T_1} \quad [6]$$

* C'den veri okumak için (DR-C'nin logik ifadesi)

$$\left. \begin{array}{l} S_1' S_0 T_2 : A \in C \\ S_1 S_0' T_2 : B \in C \end{array} \right\} \boxed{DR_C = S_1' S_0 T_2 + S_1 S_0' T_2} \quad [7]$$

* D'den veri okumak için (DR-D'nin logik ifadesi)

$$\left. \begin{array}{l} S_1' S_0' T_3 : B \in D \\ S_1' S_0 T_3 : C \in D \\ S_1 S_0' T_3 : C \in D \end{array} \right\} \boxed{DR_D = S_1' S_0' T_3 + S_1' S_0 T_3 + S_1 S_0' T_3} \quad [8]$$

[1] - [8] denklemi; kritiklerde devre ażılır!

