

Mitralieră Santinelă Automată (Sentry Gun)

Simion Ștefan 314CC

2024-12-07

Table of contents

1	Subiectul Temei	4
2	Implementare	5
2.1	Schema Bloc	5
2.2	Stări	6
2.3	Semnale Decizii	6
2.4	Semnale Ieșiri	7
3	Funcționalitate	8
4	Organigrama	10
5	Spațiul Stărilor	11
6	Tabelul tranzițiilor	12
7	Diagrame de Stare & Ecuații Rezultate	13
7.1	Q_3^{t+1}	13
7.2	Q_2^{t+1}	13
7.3	Q_1^{t+1}	14
7.4	Q_0^{t+1}	14
7.5	DAT	14
7.6	RCV	15
7.7	LOG	15
7.8	FIRE	15
8	Diagrame Karnaugh și ecuațiile rezultate pentru intrările CBB-urilor	16
8.1	Q_0	16
8.1.1	D_0	16
8.2	Q_1	18
8.2.1	J1	18
8.2.2	K1	19
8.3	Q_2	20
8.3.1	$Q_3 = 0$	21
8.3.2	$Q_3 = 1$	21

8.4	Q_3	22
8.4.1	J_3	23
8.4.2	K_3	24
9	Implementarea circuitului	26

1 Subiectul Temei

Tema proiectului constă în realizarea sintezei logice a unității de comandă al unui sentry gun (mitralieră santinelă automată). Automatul are următoarea funcționalitate:

- detectarea automată a obiectului/individului ce intră în perimetrul protejat
- poziționarea mitralierei pe obiectul/individului ce a intrat în perimetrul protejat
- împușcarea automată a obiectului/individului ce a intrat în perimetrul protejat
- primirea de instrucțiuni remote pentru controlul funcțiilor de țintire și împușcare

2 Implementare

Automatul a fost implementat pe 4 biți care codifică 4 variabile de stare: Q_3 (cel mai semnificativ), Q_2 , Q_1 , Q_0 . Variabilele de stare codifică 15 stări pe care automatul le poate avea. Cele 4 variabile de stare sunt implementate în felul următor:

- Q_0 folosind CBB tip D și un MUX 16:1.
- Q_1 folosind CBB tip JK, având implementat cu porți de tip NAND și K cu porți de tip NOR.
- Q_2 folosind CBB tip JK și un MUX 2:1.
- Q_3 folosind CBB tip JK, având J implementat printr-un MUX 4:1 și K printr-un MUX 8:1.

Ieșirile circuitului sunt implementate utilizând un decodificator 4:16, având ieșirile active pe 0.

2.1 Schema Bloc

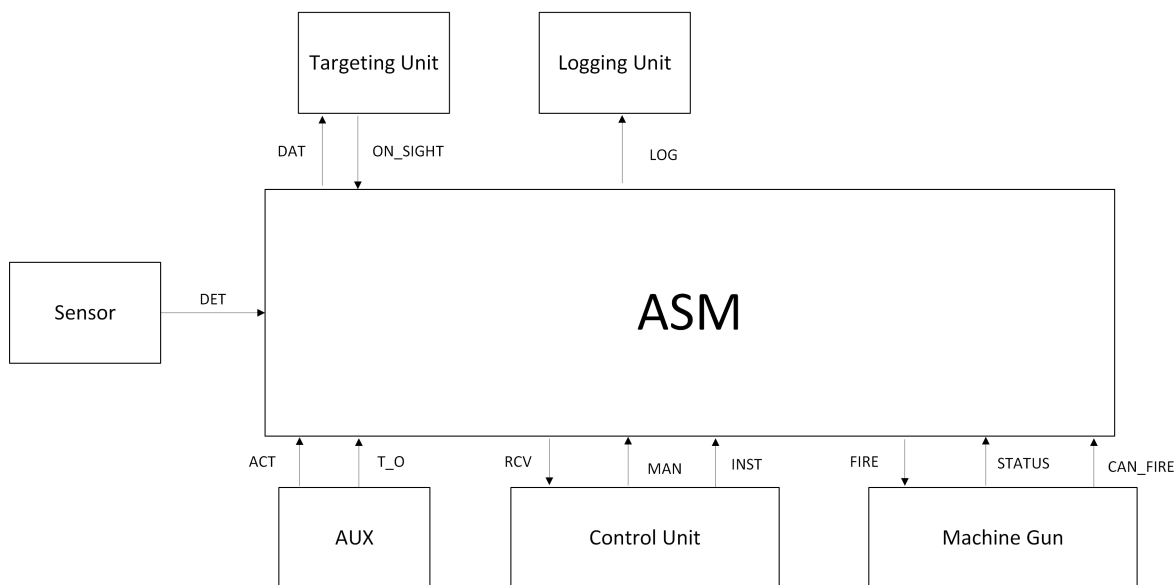


Figure 2.1: Schema Bloc Automat

2.2 Stări

- INACTIVE (0000) = Aparatul este oprit
- ACTIVE (0001) = Aparatul este activ în așteptare
- MANUAL (0101) = Aparatul funcționează manual (primește instrucțiuni manual)
- AUTO (0011) = Aparatul funcționează automat
- DETECT (0111) = Aparatul a detectat un individ/obiect în spațiul protejat
- REC_INSTR (1101) = Aparatul primește instrucțiuni remote
- CONF_INSTR (1001) = Aparatul a confirmat primirea instrucțiunilor remote
- TRANS_INFO (1111) = Aparatul a transmis datele primite de la senzor către modulul de țintire
- TRANS_INST (1010) = Aparatul a transmis instrucțiunile primite către modulul de țintire
- LOG_TARG (1110) = Aparatul a transmis informațiile de log către modulul de logging
- ON_TARG (0110) = Aparatul a pus mitraliera pe individ/obiect
- POS_FIRE (0010) = Aparatul a verificat că mitraliera poate să tragă (are muniție, etc)
- SIG_FIRE (1100) = Aparatul a încercat să pornească mitraliera
- WAIT_FIRE (0100) = Aparatul așteaptă pornirea mitralierei
- STAT_FIRE (1011) = Aparatul a împușcat individul/obiectul

2.3 Semnale Decizii

- ACT = Instrucțiunea de decizie prin care un dispozitiv auxiliar pornește sau oprește automatul
- MAN = Instrucțiunea de decizie prin care un utilizator alege dacă automatul funcționează automat sau prin instrucțiuni primite remote
- INST = Instrucțiunea de decizie prin care se verifică dacă automatul primește instrucțiuni
- DET = Instrucțiune de decizie prin care se verifică detectarea unui individ/obiect în zona protejată
- ON_SIGHT = Instrucțiunea de decizie prin care se verifică poziționarea mitralierei pe individ/obiect
- CAN_FIRE = Instrucțiunea de decizie prin care se verifică dacă mitraliera este funcțională
- STATUS = Instrucțiunea de decizie prin care se verifică dacă mitraliera a început să tragă
- T_O = Instrucțiunea de decizie (timer) prin care se verifică dacă a trecut perioada de timp în care trebuia să se pornească mitraliera (time_out)

2.4 Semnale leşiri

- DAT = Automatul transmite informațiile primite de la senzor către modulul de țintire
- RCV = Automatul primește instrucțiunile trimise remote
- FIRE = Automatul trimite semnalul de foc/tragere către mitralieră
- LOG = Automatul trimite informații relevante către unitatea de logging

3 Funcționalitate

Automatul pornește în starea **INACTIVE** în care este oprit. Acesta se activează când primește semnalul de pornire (**ACT**) și ajunge în starea **ACTIVE**.

Dacă este selectat modul de funcționare manual prin semnalul **MAN**, atunci automatul intră în starea **MANUAL**. Dacă acesta nu primește instrucțiuni atunci acesta se reîntoarce în starea **INACTIVE**. Dacă primește instrucțiuni (**INST**) automatul trece în starea **REC_INSTR**, apoi trimite semnalul **RCV**, confirmând că a primit instrucțiunile. Automatul trece în starea **CONF_INSTR**, și trimite semnalul **DAT**, semnalând că trimite instrucțiunile primite către modulul de țintire. Apoi trece în starea **TRANS_INST** și trimite semnalul **LOG** indicând că trimite informații de log (instrucțiunile primite și pașii îndepliniți pentru a le urma) către modulul de logging și trece în starea **ON_TARG**, mitraliera fiind poziționată pe locația dorită.

Dacă nu a fost selectat modul de funcționare manual, sentry gun-ul intră în modul automat, trecând în starea **AUTO**. Dacă senzorul detectează un individ/obiect în spațiul protejat aparatul trece în starea **DETECT**, fiind conștientizată prezența unui individ/obiect, altfel în **INACTIVE**. Din **DETECT**, automatul trimite semnalul **DAT** semnalând că trimite datele primite de la senzor către modulul de țintire și trece în starea **TRANS_INFO**. Apoi acesta transmite semnalul **LOG**, indicând că transmite informațiile de log (informațiile primite de la senzor și pașii îndepliniți pentru a ținti mitraliera pe individ/obiect) către modulul de logging și trece în starea **LOG_TARG**. Dacă mitraliera este poziționată corect pe individ/obiect primește semnalul **ON_SIGHT** și trece în starea **ON_TARG**, aparatul știind că are mitraliera poziționată corect pe individ/obiect, altfel trece în starea **AUTO**.

Din starea **ON_TARG** automatul verifică dacă mitraliera este funcțională (are muniție, etc) prin semnalul **CAN_FIRE**. Dacă mitraliera nu poate trage, automatul trece în starea **INACTIVE**, iar dacă aceasta este funcțională trece în starea **POS_FIRE**, aparatul având posibilitatea de a trage în locul dorit sau în individul/obiectul care a intrat în spațiul protejat. Având posibilitatea de a trage, automatul trimite semnalul **FIRE** către mitraliera pentru a o porni și trece în starea **SIG_FIRE**. Apoi automatul verifică dacă mitraliera a început să tragă prin semnalul **STATUS**.

Dacă mitraliera nu a început să tragă, atunci automatul trece în starea **WAIT_FIRE**, aparatul așteptând să pornească mitraliera. Automatul verifică dacă a trecut prea mult timp de când a trimis primul semnal de **FIRE** (= dacă a făcut timeout), prin semnalul **T_0**. Dacă da, atunci trece în starea **INACTIVE**, iar dacă nu trece în starea **POS_FIRE**.

Dacă mitraliera pornește (automatul primește semnalul **STATUS**), atunci automatul trece în starea **STAT_FIRE**, aparatul împușcând locația dorită sau individul/obiectul care a intrat în

spațiul protejat și trimite semnalul LOG, indicând că transmite informații de log (cât a durat să se pornească mitraliera, muniția consumată, etc), către modulul de logging și trece în starea **ACTIVE**.

4 Organigrama

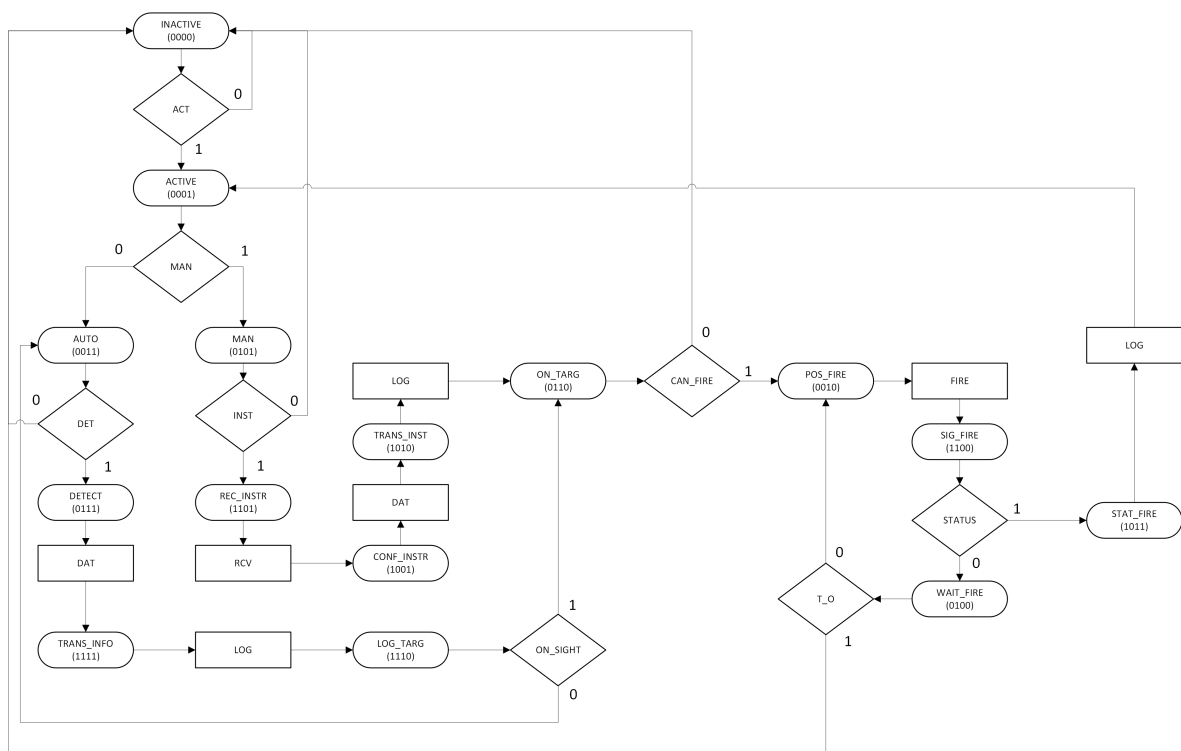


Figure 4.1: Organigrama Automatului

5 Spațiul Stărilor

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	INACTIVE	ACTIVE	AUTO	POS_FIRE
01	WAIT_FIRE	MANUAL	DETECT	ON_TARG
11	SIG_FIRE	REC_INSTR	TRANS_INFO	LOG_TARG
10	*	CONF_INSTR	STAT_FIRE	TRANS_INST

6 Tabelul tranzițiilor

Q_3^t	Q_2^t	Q_1^t	Q_0^t	Q_3^{t+1}	Q_2^{t+1}	Q_1^{t+1}	Q_0^{t+1}	Exits
0	0	0	0	0	0	0	ACT	0
0	0	0	1	0	MAN	\overline{MAN}	1	0
0	0	1	0	1	1	0	0	FIRE
0	0	1	1	0	DET	\overline{DET}	DET	0
0	1	0	0	0	0	$\overline{T_O}$	0	0
0	1	0	1	INST	INST	0	INST	0
0	1	1	0	0	0	CAN_FIRE	0	0
0	1	1	1	1	1	1	1	DAT
1	0	0	0	*	*	*	*	*
1	0	0	1	1	0	1	0	DAT
1	0	1	0	0	1	1	0	LOG
1	0	1	1	0	0	0	1	LOG
1	1	0	0	STATUS	\overline{STATUS}	STATUS	STATUS	0
1	1	0	1	1	0	0	1	RCV
1	1	1	0	0	ON_SIGHT	1	$\overline{ON_SIGHT}$	0
1	1	1	1	1	1	1	0	LOG

7 Diagrame de Stare & Ecuații Rezultate

7.1 Q_3^{t+1}

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	0	<u>1</u>
01	0	INST	<u>1</u>	0
11	STATUS	<u>1</u>	<u>1</u>	0
10	*	<u>1</u>	0	0

$$Q_3^{t+1} = \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} + Q_2 Q_1 Q_0 + Q_3 \overline{Q_1} Q_0 + \text{STATUS} Q_3 \overline{Q_1} + \text{INST} Q_2 Q_0$$

7.2 Q_2^{t+1}

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	MAN	DET	<u>1</u>
01	0	INST	<u>1</u>	0
11	STATUS	0	<u>1</u>	ON_SIGHT
10	*	0	0	<u>1</u>

$$Q_2^{t+1} = \overline{Q_2} Q_1 \overline{Q_0} + Q_2 Q_1 Q_0 + \text{ON_SIGHT} Q_3 Q_1 \overline{Q_0} + \text{MAN} \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 + \text{DET} \overline{Q_3} Q_1 Q_0 + \text{INST} \overline{Q_3} Q_2 Q_0 + \text{STATUS} Q_3 \overline{Q_1} \overline{Q_0}$$

7.3 Q_1^{t+1}

$Q_3 Q_2 \setminus Q_1 Q_0$	00	01	11	10
00	0	MAN	DET	0
01	T O	0	1	CAN_FIRE
11	STATUS	0	1	1
10	*	1	0	1

$$Q_1^{t+1} = Q_3 Q_1 \overline{Q_0} + Q_2 Q_1 Q_0 + Q_3 \overline{Q_2} \overline{Q_1} + \overline{T} \overline{O} \overline{Q_3} Q_2 \overline{Q_1} \overline{Q_0} + \text{STATUS} Q_3 \overline{Q_1} \overline{Q_0} + \overline{\text{MAN}} \overline{Q_2} \overline{Q_1} Q_0 + \text{DET} \overline{Q_3} Q_1 Q_0 + \text{CAN_FIRE} Q_2 Q_1$$

7.4 Q_0^{t+1}

$Q_3 Q_2 \setminus Q_1 Q_0$	00	01	11	10
00	ACT	1	DET	0
01	0	INST	1	0
11	STATUS	1	0	ON_SIGHT
10	*	0	1	0

$$Q_0^{t+1} = Q_3 \overline{Q_2} Q_1 Q_0 + \overline{Q_3} Q_2 \overline{Q_1} Q_0 + \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 + Q_3 Q_2 \overline{Q_1} Q_0 + \overline{\text{ON_SIGHT}} Q_3 Q_2 Q_1 \overline{Q_0} + \text{DET} \overline{Q_3} Q_1 Q_0 + \text{ACT} \overline{Q_3} \overline{Q_2} \overline{Q_1} + \text{INST} Q_2 \overline{Q_1} Q_0 + \text{STATUS} Q_3 \overline{Q_1} \overline{Q_0}$$

7.5 DAT

$Q_3 Q_2 \setminus Q_1 Q_0$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	0
10	*	1	0	0

$$\text{DAT} = \overline{Q_3} Q_2 Q_1 Q_0 + Q_3 \overline{Q_2} \overline{Q_1}$$

7.6 RCV

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	1	0	0
10	*	0	0	0

$$\text{RCV} = Q_3Q_2\overline{Q_1}Q_0$$

7.7 LOG

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	*	0	1	1

$$\text{LOG} = Q_3Q_1Q_0 + Q_3\overline{Q_2}Q_1$$

7.8 FIRE

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	0	0	0	0
10	*	0	0	0

$$\text{FIRE} = \overline{Q_3}\overline{Q_2}Q_1\overline{Q_0}$$

8 Diagrame Karnaugh și ecuațiile rezultate pentru intrările CBB-urilor

Mențiune: În acest capitol am adăugat părți ale circuitului complet în cadrul fiecărui subcapitol, pentru a ușura vizualizarea lor, însă acestea diferă puțin de implementarea lor actuală. Circuitul complet utilizează magistrale pentru $\overline{Q_i}$, $i = \overline{0, 3}$, dar secvențele din subcapitole utilizează porți logice pentru implementarea lor (porți NOR/NAND în capitolele în care se cere implementarea cu acestea și porți NOT în restul), pentru a demonstra înțelegerea acestora.

8.1 Q_0

Implementat printr-un CBB tip D și un Mux 16:1

8.1.1 D_0

$Q_3 Q_2 \setminus Q_1 Q_0$	00	01	11	10
00	ACT	1	DET	0
01	0	INST	1	0
11	STATUS	1	0	$\overline{\text{ON_SIGHT}}$
10	*	0	1	0

$$D_0 = Q_3 \overline{Q_2} Q_1 Q_0 + \overline{Q_3} Q_2 Q_1 Q_0 + \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 + Q_3 Q_2 \overline{Q_1} Q_0 + \overline{\text{ON_SIGHT}} Q_3 Q_2 Q_1 \overline{Q_0} + \text{DET} \overline{Q_3} Q_1 Q_0 + \text{ACT} \overline{Q_3} \overline{Q_2} \overline{Q_1} + \text{INST} Q_2 \overline{Q_1} Q_0 + \text{STATUS} Q_3 \overline{Q_1} \overline{Q_0}$$

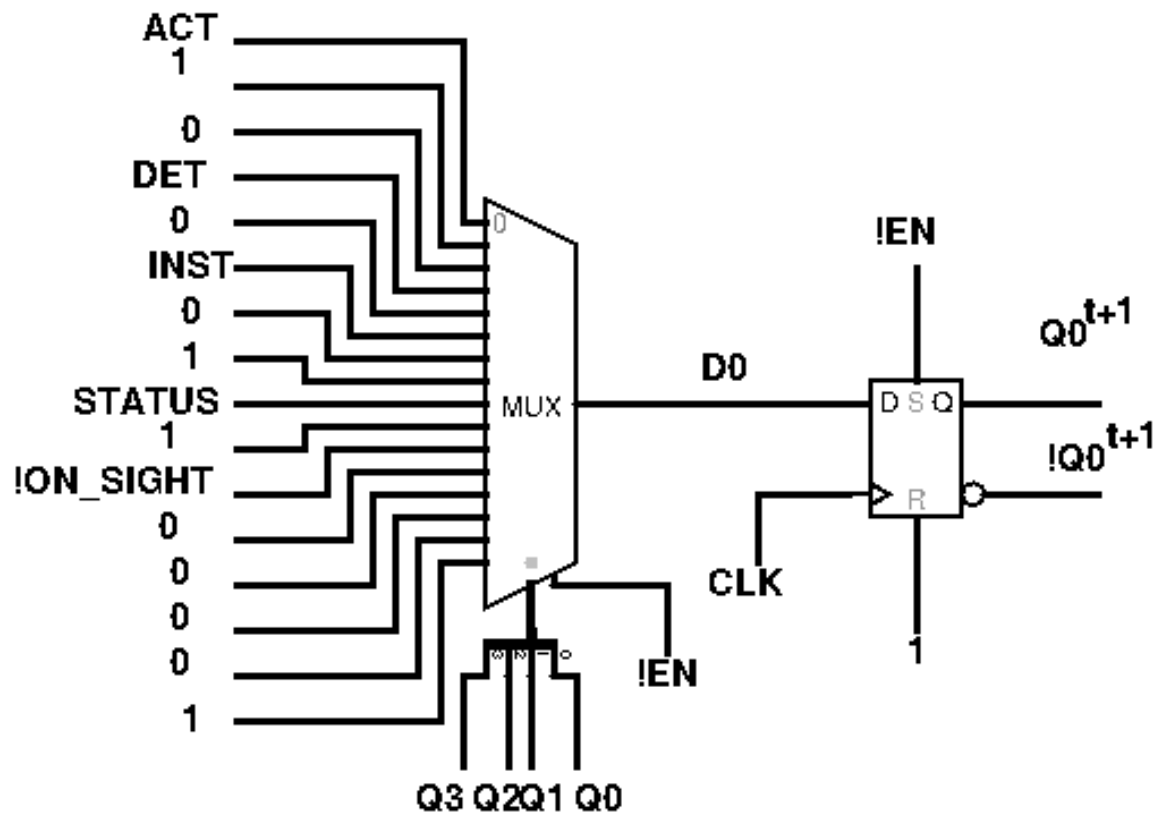


Figure 8.1: Implementare Q_0

8.2 Q₁

Implementat printr-un CBB de tip JK; J implementat cu porți NAND și K implementat cu porți NOR

Q_1^t	Q_1^{t+1}	J_1	K_1
0	0	0	*
0	\overline{MAN}	\overline{MAN}	*
1	0	*	1
1	DET	*	\overline{DET}
0	$\overline{T_O}$	$\overline{T_O}$	*
0	0	0	*
1	CAN_FIRE	*	$\overline{CAN_FIRE}$
1	1	*	0
0	*	*	*
0	1	1	*
1	1	*	0
1	0	*	1
0	STATUS	STATUS	*
0	0	0	*
1	1	*	0
1	1	*	0

8.2.1 J1

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	0	\overline{MAN}	*	*
01	$\overline{T_O}$	0	*	*
11	STATUS	0	*	*
10	*	1	*	*

$$J_1 = Q_3\overline{Q_2} + \overline{MAN}\overline{Q_2}Q_0 + STATUSQ_3\overline{Q_0} + \overline{T_O}\overline{Q_3}Q_2\overline{Q_0}$$

Pentru a implementa cu porți NAND am negat de 2 ori J_1 .

$$J_1 = \overline{\overline{Q_3\overline{Q_2}}\overline{\overline{MAN}\overline{Q_2}Q_0}}\overline{\overline{STATUSQ_3\overline{Q_0}}\overline{\overline{T_O}\overline{Q_3}Q_2\overline{Q_0}}}$$

8.2.2 K1

$Q_3 Q_2 \setminus Q_1 Q_0$	00	01	11	10
00	*	*	DET	1
01	*	*	0	CAN_FIRE
11	*	*	0	0
10	*	*	1	0

$$K_1 = \overline{Q_3} \overline{Q_2} \overline{Q_0} + Q_3 \overline{Q_2} Q_0 + \overline{\text{DET}} \overline{Q_3} \overline{Q_2} + \overline{\text{CAN_FIRE}} \overline{Q_3} \overline{Q_0}$$

Pentru a implementa cu porți NOR am negat de 2 ori fiecare minterm

$$K_1 = \overline{Q_3 + Q_2 + Q_0} + \overline{\overline{Q_3} + \overline{Q_2} + \overline{Q_0}} + \overline{\text{DET} + Q_3 + Q_2} + \overline{\text{CAN_FIRE} + Q_3 + Q_0}$$

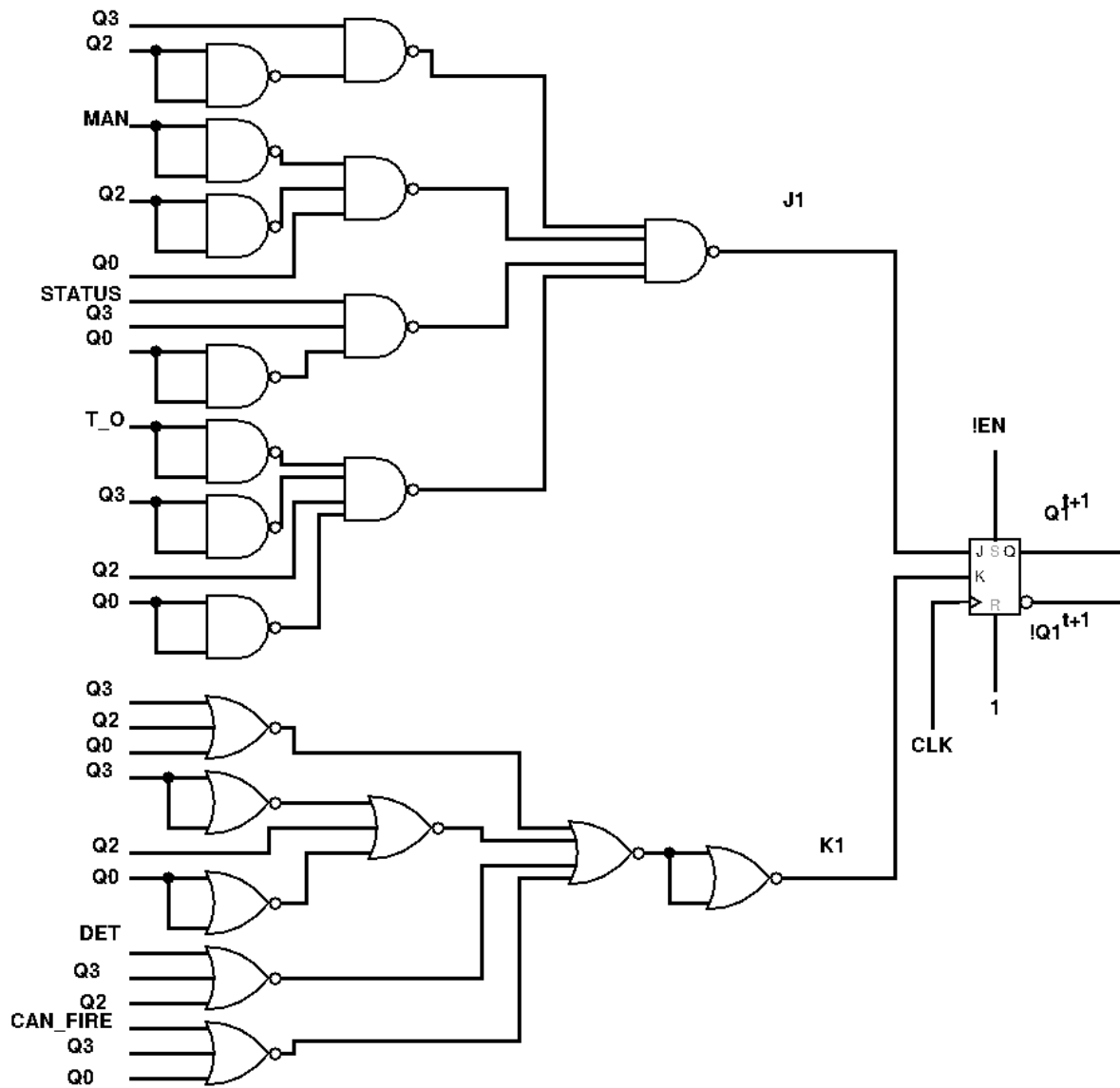


Figure 8.2: Implementare Q1

8.3 Q_2

Implementat printr-un CBB de tip D și un Mux 2:1.

Am ales Q_3 ca bit de selectie.

8.3.1 $Q_3 = 0$

$Q_2 \setminus Q_1 Q_0$	00	01	11	10
0	0	MAN	DET	(1)
1	*	INST	(1)	0

Intrarea 0: $Q_2 Q_1 Q_0 + \overline{Q_2} Q_1 \overline{Q_0} + \text{DET} \overline{Q_2} Q_1 + \text{MAN} \overline{Q_2} \overline{Q_1} Q_0 + \text{INST} Q_2 Q_0$

8.3.2 $Q_3 = 1$

$Q_2 \setminus Q_1 Q_0$	00	01	11	10
0	STATUS	0	①	ON SIGHT
1	*	0	0	1

Intrarea 1: $\overline{Q_2} Q_1 Q_0 + Q_2 \overline{Q_0} + \overline{\text{STATUS}} \overline{Q_1} \overline{Q_0} + \text{ON_SIGHT} Q_1 \overline{Q_0}$

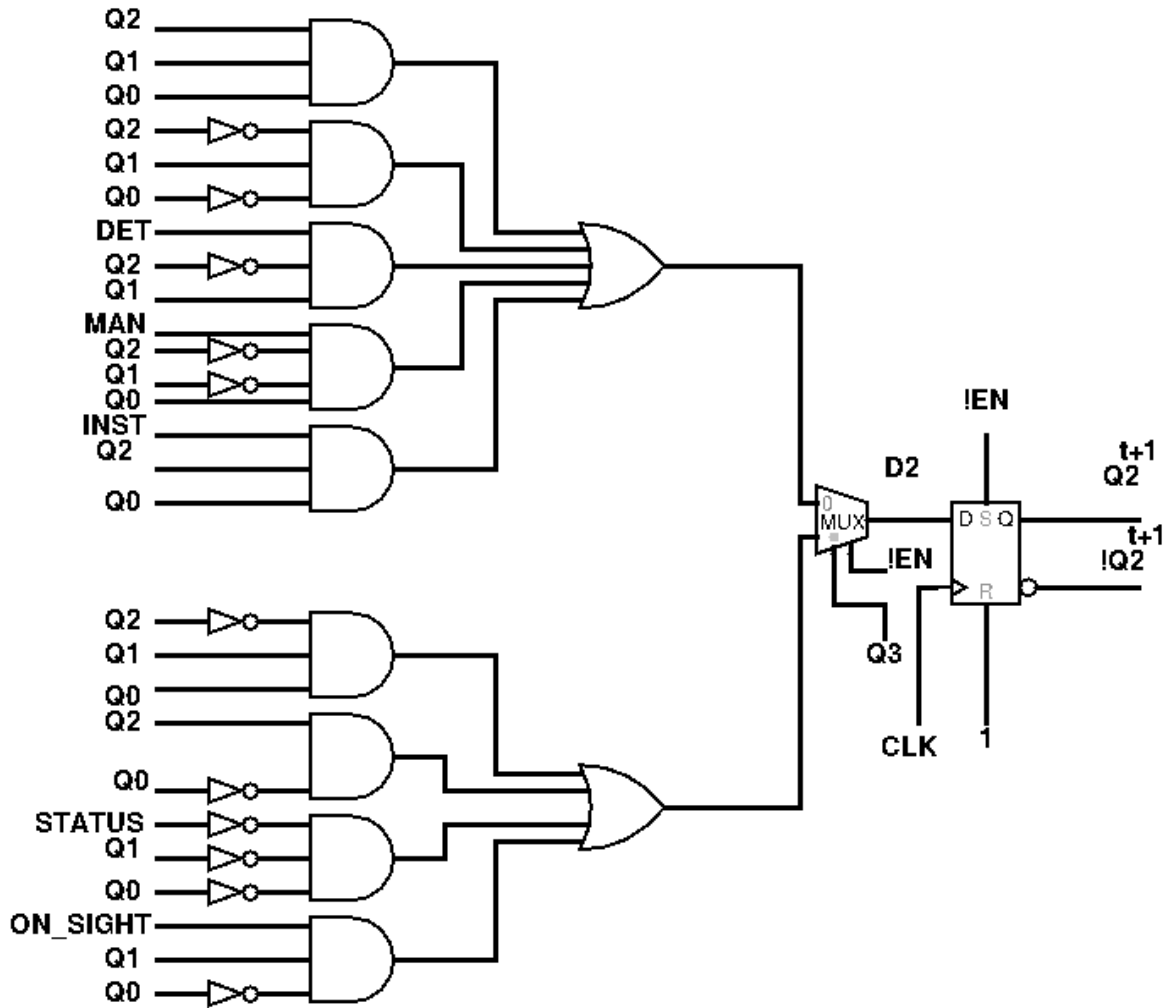


Figure 8.3: Implementare Q2

8.4 Q₃

Implementat printr-un CBB de tip JK; J este implementat printr-un Mux 4:1 și K este implementat printr-un MUX 8:1

8.4.1 J₃

$Q_3 Q_2 \setminus Q_1 Q_0$	00	01	11	10
00	0	0	0	1
01	0	INST	1	0
11	*	*	*	*
10	*	*	*	*

Am ales ca variabile de selecție Q_3 și Q_2 .

8.4.1.1 $Q_3 Q_2 = 00$

$Q_1 \setminus Q_2$	0	1
0	0	0
1	①	0

Intrarea 00: $Q_1 \overline{Q_0}$

8.4.1.2 $Q_3 Q_2 = 01$

$Q_1 \setminus Q_2$	0	1
0	0	INST
1	0	①

Intrarea 01: $Q_1 Q_0 + \text{INST} Q_0$

8.4.1.3 $Q_3 Q_2 = 10$

$Q_1 \setminus Q_2$	0	1
0	*	*
1	*	*

Intrarea 10: 0

8.4.1.4 $Q_3Q_2 = 11$

$Q_1 \setminus Q_2$	0	1
0	*	*
1	*	*

Intrarea: 11: 0

8.4.2 K_3

$Q_3Q_2 \setminus Q_1Q_0$	00	01	11	10
00	*	*	*	*
01	*	*	*	*
11	STATUS	0	0	1
10	*	0	1	1

Am ales ca variabile de selecție Q_3 , Q_2 și Q_0 .

8.4.2.1 $Q_3Q_2Q_1 = 000$

Intrarea 000: 0

8.4.2.2 $Q_3Q_2Q_1 = 001$

Intrarea 001: 0

8.4.2.3 $Q_3Q_2Q_1 = 010$

Intrarea 010: 0

8.4.2.4 $Q_3Q_2Q_1 = 011$

Intrarea 011: 0

8.4.2.5 $Q_3Q_2Q_1 = 100$

Intrarea 100: 0

8.4.2.6 $Q_3Q_2Q_1 = 101$

Intrarea 101: 1

8.4.2.7 $Q_3Q_2Q_1 = 110$

Intrarea 110: $\overline{\text{STATUS}} \overline{Q_0}$

8.4.2.8 $Q_3Q_2Q_1 = 111$

Intrarea 111: $\overline{Q_0}$

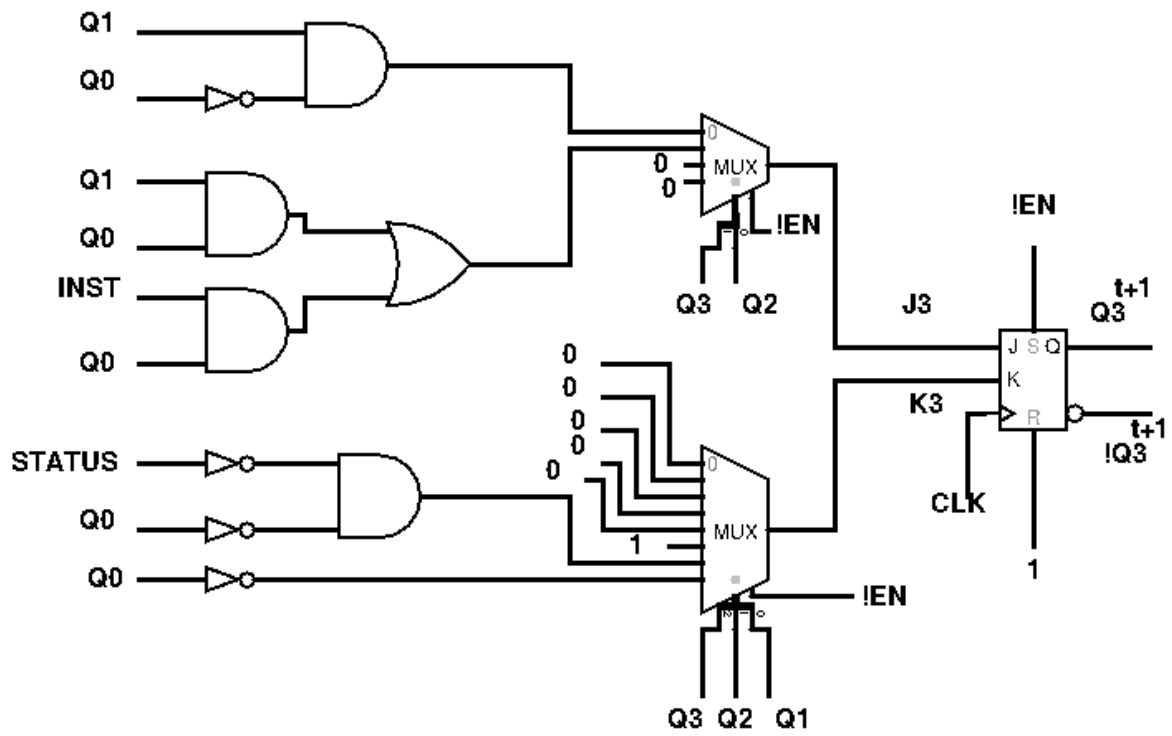


Figure 8.4: Implementare Q_3

9 Implementarea circuitului

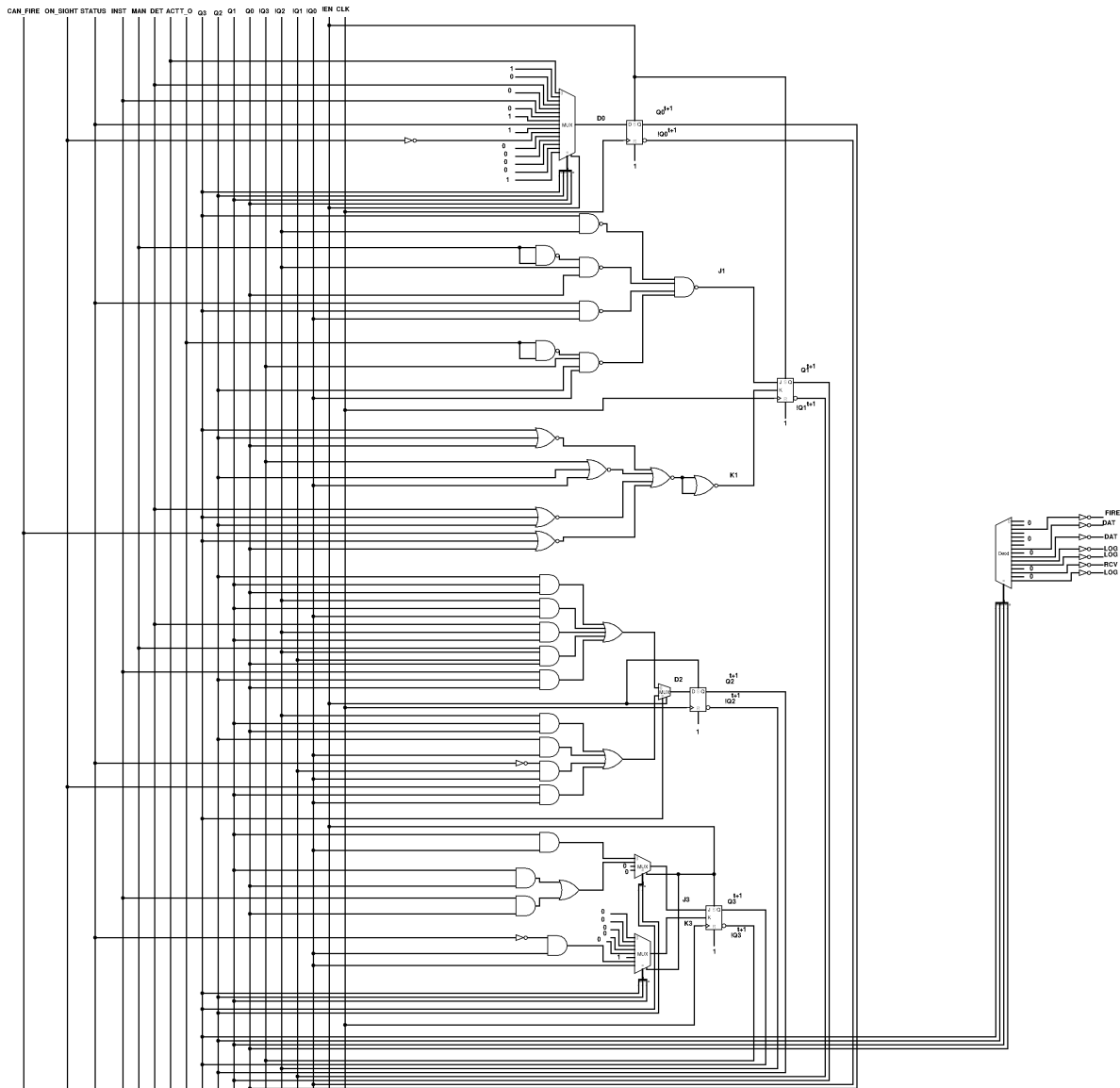


Figure 9.1: Circuit