

Exempel på RTN-beskrivning av FLEX-instruktioner

1. Figur 1 på sidan 12 i detta häfte visar hur datorn FLEX är uppbyggd. På sidan 11 visas dessutom hur ALU:ns funktion väljs med styrsignalerna $f_3 - f_0$ och hur C_{in} väljs med styrsignalerna g_1 och g_0 .

- a) I tabellen nedan visas utförandefasen för en av FLEX-processorns instruktioner. Vilken instruktion har denna EXECUTE-sekvens? Skriv instruktionen med assemblerspråk.

State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	M+1→R, Flags→CC	MR, f_3 , g_0 , LD _R , LD _{CC} .
Q ₈	R→M	OE _R , MW, (NF).

- b) Rita en tabell motsvarande den i a) ovan, som visar utförandefasen för maskininstruktionen ANDA Adr för FLEX-processorn. I instruktionslistan för FLEX-processorn beskrivs instruktionen enligt följande tabell.

Instruktion		Adressering			Operationsbeskrivning	Flaggpåverkan			
Operation	Beteckning	Absolute				3	2	1	0
		OP	#	~		N	Z	V	C
AND	ANDA Adr	17	2	7	A AND M(Adr) → A	a	a	0	0

I tabellen anges antalet states till 7, varav 2 tillhör hämtfasen.

2. Figur 1 på sidan 12 i detta häfte visar hur datorn FLEX är uppbyggd. På sidan 11 visas dessutom hur ALU:ns funktion väljs med styrsignalerna $f_3 - f_0$ och hur C_{in} väljs med styrsignalerna g_1 och g_0 .

I tabellen nedan visas styrsignalerna för utförandefasen för en av FLEX-processorns instruktioner. Dessutom visas RTN-beskrivningen för tillstånd 0 i EXECUTE-sekvensen.

- a) Ge RTN-beskrivningen för de övriga tillstånden. Använd samma skrivsätt som i RTN-beskrivningen för den första raden.
- b) Vilken instruktion har denna EXECUTE-sekvens? Skriv instruktionen med assemblerspråk.

State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆		MR, LD _{MA} , LD _R .
Q ₇		MR, LD _T .
Q ₈		OE _R , f_3 , f_2 , g_0 , LD _R , LD _{CC} .
Q ₉		OE _R , MW, (NF).

- c) Rita en tabell motsvarande den ovan, som visar utförandefasen för maskininstruktionen JMP Adr för FLEX-processorn. I instruktionslistan för FLEX-processorn beskrivs instruktionen enligt följande tabell.

Instruktion		Adressering			Operationsbeskrivning	Flaggpåverkan			
Operation	Beteckning	Absolute				3	2	1	0
		OP	#	~		N	Z	V	C
Unconditional jump	JMP Adr	59	2	4	Adr → PC	•	•	•	•

I tabellen anges antalet tillstånd till 4, varav 2 tillhör FETCH-sekvensen.

3. Figur 1 på sidan 12 i detta häfte visar hur datorn FLEX är uppbyggd. På sidan 11 visas dessutom hur ALU:ns funktion väljs med styrsignalerna $f_3 - f_0$ och hur C_{in} väljs med styrsignalerna g_1 och g_0 .

I tabellen nedan visas styrsignalerna för utförandefasen för en av FLEX-processorns instruktioner. Dessutom visas RTN-beskrivningen för tillstånd 0 i EXECUTE-sekvensen.

- a) Ge RTN-beskrivningen för de övriga tillstånden. Använd samma skrivsätt som i RTN-beskrivningen för den första raden.
- b) Vilken instruktion har denna "execute"-fas? Skriv instruktionen med assemblerspråk.

State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆		MR, LD _{MA} .
Q ₇		MR, LD _T .
Q ₈		OE _A , f_3, f_1 , LD _R , LD _{CC} .
Q ₉		OE _R , LD _A , (NF).

- c) Rita en tabell motsvarande den ovan, som visar utförandefasen för maskininstruktionen BRA Adr för FLEX-processorn. I instruktionslistan för FLEX-processorn beskrivs instruktionen enligt följande tabell.

Instruktion		Adressering			Operations-beskrivning	Flagg-påverkan			
Operation	Beteckning	Relative				3	2	1	0
		OP	#	~		N	Z	V	C
Unconditional branch	BRA Adr	5A	2	5	PC+Offs → PC	•	•	•	•

I tabellen anges antalet states till 5, varav 2 tillhör hämtfasen.

Lösningar

1. a) I state Q_5 kopieras innehållet i PC till MA-registret. Detta innebär att adressen till minnesordet efter OP-koden hamnar i MA-registret. PC ökas med ett så att den pekar på nästa operationskod i minnet.

I state Q_6 läses minnesordet på adressen efter OP-koden och placeras i MA-registret. Detta innebär att instruktionens adressdel placeras i MA.

I state Q_7 läses minnesinnehållet som ökas med ett och placeras i R. Flaggorna uppdateras.

I state Q_8 skrivs det ökade minnesinnehållet tillbaka i minnet på den adress som finns i MA, dvs instruktionens adressdel och executesekvensen avslutas.

Detta innebär att instruktionen är en **INC Adr**

- b) **ANDA Adr** Ord1: 17H(Opkod) Ord2: Adr

State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q_5	$PC \rightarrow MA, PC+1 \rightarrow PC$	$OE_{PC}, LD_{MA}, Inc_{PC}$.
Q_6	$M \rightarrow MA$	MR, LD_{MA} .
Q_7	$M \rightarrow T$	MR, LD_T .
Q_8	$A \text{ AND } T \rightarrow R, \text{Flags} \rightarrow CC$	$OE_A, f_2, f_1, LD_R, LD_{CC}$.
Q_9	$R \rightarrow A$	$OE_R, LD_A, (NF)$.

(Efter det sista tillståndet i EXECUTE följer det första tillståndet i FETCH.)

2. a)

State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA, 00H→R	MR, LD _{MA} , LD _R .
Q ₇	M→T	MR, LD _T .
Q ₈	R – T→R, Flags→CC	OE _R , f ₃ , f ₂ , g ₀ , LD _R , LD _{CC} .
Q ₉	R→M	OE _R , MW, (NF).

(Efter det sista tillståndet i EXECUTE följer det första tillståndet i FETCH.)

- b) I state Q₅ kopieras innehållet i PC till MA-registret. Detta innebär att adressen till minnesordet efter OP-koden hamnar i MA-registret. PC ökas med ett så att den pekar på nästa operationskod i minnet.

I state Q₆ läses minnesordet på adressen efter OP-koden och placeras i MA-registret. Detta innebär att instruktionens adressdel placeras i MA. Dessutom placeras värdet 00H i R-registret.

I state Q₇ läses minnesinnehållet som pekas ut av instruktionens adressdel och placeras i T-registret.

I state Q₈ bildas skillnaden $0 - M(\text{Adr}) = -M(\text{Adr})$ som placeras i R-registret. Flaggor uppdateras.

I state Q₉ skrivs $-M(\text{Adr})$ tillbaka i minnet, dvs $-M(\text{Adr}) \rightarrow M(\text{Adr})$, och executesekvensen avslutas.

Detta innebär att instruktionen är en **NEG Adr**

c)

JMP Adr

Ord1: 59H(Opkod) Ord2: Adr

State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	PC→MA	OE _{PC} , LD _{MA} .
Q ₆	M→PC	MR, LD _{PC} , (NF).

(Efter det sista tillståndet i EXECUTE följer det första tillståndet i FETCH.)

3. a)

State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	M→T	MR, LD _T .
Q ₈	A + T→R, Flags →CC	OE _A , f ₃ , f ₁ , LD _R , LD _{CC} .
Q ₉	R→A	OE _R , LD _A , (NF).

(Efter det sista tillståndet i EXECUTE följer det första tillståndet i FETCH.)

b)

I state Q₅ kopieras innehållet i PC till MA-registret. Detta innebär att adressen till minnesordet efter OP-koden hamnar i MA-registret. PC ökas med ett så att den pekar på nästa operationskod i minnet.

I state Q₆ läses minnesordet på adressen efter OP-koden och placeras i MA-registret. Detta innebär att instruktionens adressdel placeras i MA.

I state Q₇ läses minnesinnehållet som pekas ut av instruktionens adressdel och placeras i T-registret.

I state Q₈ bildas summan A + M(Adr) som placeras i R-registret. Flaggorna uppdateras.

I state Q₉ kopieras summan från R-registret till A-registret och executesekvensen avslutas.

Detta innebär att instruktionen är en **ADDA Adr**

c) **BRA Adr**

Ord1: 5AH(Opkod) Ord2: Offset

CP	State	RTN-beskrivning	Aktiva styrsignaler (=1)
Q ₅	0	PC→MA, PC→T	OE _{PC} , LD _{MA} , LD _T .
Q ₆	1	M+T+1→R	MR, f ₃ , f ₁ , g ₀ , LD _R .
Q ₇	2	R→PC	OE _R , LD _{PC} , (NF).

(Efter det sista tillståndet i EXECUTE följer det första tillståndet i FETCH.)

Beskrivning av EXECUTE-fasen för vissa FLEX-instruktioner (Efter det sista tillståndet i EXECUTE följer det första tillståndet i FETCH.)

Observera att dessa RTN-beskrivningar inte gäller för FLISP!

NOP Ord1: 00H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅		(NF).

TFR A,B Ord1: 01H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	A→B	OE _A , LD _B , (NF).

TFR A,CC Ord1: 03H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	A→CC	OE _A , g ₂ , LD _{CC} , (NF).

EXG A,B Ord1: 07H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	A→R	OE _A , f ₀ , LD _R .
Q ₆	B→A	OE _B , LD _A .
Q ₇	R→B	OE _R , LD _B , (NF).

LDA #Data Ord1: 0FH(Opkod) Ord2: Data

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→A	MR, LD _A , (NF).

STAA Adr Ord1: 13H(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	A→M	OE _A , MW, (NF).

ANDA #Data Ord1: 19H(Opkod) Ord2: Data

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→T	MR, LD _T .
Q ₇	A AND T→R, Flaggor→CC	OE _A , f ₂ , f ₁ , LD _R , LD _{CC} .
Q ₈	R→A	OE _R , LD _A , (NF).

ORAB Adr Ord1: 1CH(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	M→T	MR, LD _T .
Q ₈	B OR T→R, Flaggor→CC	OE _B , f ₂ , f ₀ , LD _R , LD _{CC} .
Q ₉	R→B	OE _R , LD _B , (NF).

COMA

Ord1: 23H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	A'→R, Flaggor→CC	OE _A , f ₁ , f ₀ , LD _R , LD _{CC} .
Q ₆	R→A	OE _R , LD _A , (NF).

ANDCC #Data

Ord1: 26H(Opkod) Ord2: Data

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→T	MR, LD _T .
Q ₇	CC AND T→R	OE _{CC} , f ₂ , f ₁ , LD _R .
Q ₈	R→CC	OE _R , LD _{CC} , g ₂ , (NF).

ADDB Adr

Ord1: 29H(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	M→T	MR, LD _T .
Q ₈	B+T→R, Flaggor→CC	OE _B , f ₃ , f ₁ , LD _R , LD _{CC} .
Q ₉	R→B	OE _R , LD _B , (NF).

ADCA #Data

Ord1: 2EH(Opkod) Ord2: Data

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→T	MR, LD _T .
Q ₇	A+T+C→R, Flaggor→CC	OE _A , f ₃ , f ₁ , g ₁ , LD _R , LD _{CC} .
Q ₈	R→A	OE _R , LD _A , (NF).

SUBA Adr

Ord1: 30H(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	M→T	MR, LD _T .
Q ₈	A-T→R, Flaggor→CC	OE _A , f ₃ , f ₂ , g ₀ , LD _R , LD _{CC} .
Q ₉	R→A	OE _R , LD _A , (NF).

SBCA Adr

Ord1: 34H(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	M→T	MR, LD _T .
Q ₈	A-T-C→R, Flaggor→CC	OE _A , f ₃ , f ₂ , g ₁ , g ₀ , LD _R , LD _{CC} .
Q ₉	R→A	OE _R , LD _A , (NF).

NEG Adr

Ord1: 3AH(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA, 00H→R	MR, LD _{MA} , LD _R .
Q ₇	M→T	MR, LD _T .
Q ₈	R-T→R, Flaggor→CC	OE _R , f ₃ , f ₂ , g ₀ , LD _R , LD _{CC} .
Q ₉	R→M	OE _R , MW, (NF).

ASLA

Ord1: 3BH(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	2A→R, Flaggor→CC	OE _A , f ₃ , f ₁ , f ₀ , LD _R , LD _{CC} .
Q ₆	R→A	OE _R , LD _A , (NF).

ASL Adr

Ord1: 3DH(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	2M→R, Flaggor→CC	MR, f ₃ , f ₁ , f ₀ , LD _R , LD _{CC} .
Q ₈	R→M	OE _R , MW, (NF).

ROLA

Ord1: 3EH(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	2A+C→R, Flaggor→CC	OE _A , f ₃ , f ₁ , f ₀ , g ₁ , LD _R , LD _{CC} .
Q ₆	R→A	OE _R , LD _A , (NF).

ROL Adr

Ord1: 40H(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	2M+C→R, Flaggor→CC	MR, f ₃ , f ₁ , f ₀ , g ₁ , LD _R , LD _{CC} .
Q ₈	R→M	OE _R , MW, (NF).

INCA

Ord1: 41H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	A+1→R, Flaggor→CC	OE _A , f ₃ , g ₀ , LD _R , LD _{CC} .
Q ₆	R→A	OE _R , LD _A , (NF).

CLRA

Ord1: 47H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	00H→R, Flaggor→CC	LD _R , LD _{CC} .
Q ₆	R→A	OE _R , LD _A , (NF).

CMPA #Data Ord1: 4EH(Opkod) Ord2: Data

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→T	MR, LD _T .
Q ₇	A-T, Flaggor→CC	OE _A , f ₃ , f ₂ , g ₀ , LD _{CC} , (NF).

TSTA Ord1: 52H(Opkod)

State	RTN-beskrivning	Styrsignaler
Q ₅	A→R, Flaggor→CC	OE _A , LD _R , f ₀ , LD _{CC} , (NF).

BITA Adr Ord1: 55H(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC+1→PC	OE _{PC} , LD _{MA} , IncPC.
Q ₆	M→MA	MR, LD _{MA} .
Q ₇	M→T	MR, LD _T .
Q ₈	A AND T, Flaggor→CC	OE _A , f ₂ , f ₁ , LD _{CC} , (NF).

JMP Adr Ord1: 59H(Opkod) Ord2: Adr

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA	OE _{PC} , LD _{MA} .
Q ₆	M→PC	MR, LD _{PC} , (NF).

BRA Adr Ord1: 5AH(Opkod) Ord2: Offset

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC→T	OE _{PC} , LD _{MA} , LD _T .
Q ₆	M+T+1→R	MR, f ₃ , f ₁ , g ₀ , LD _R .
Q ₇	R→PC	OE _R , LD _{PC} , (NF).

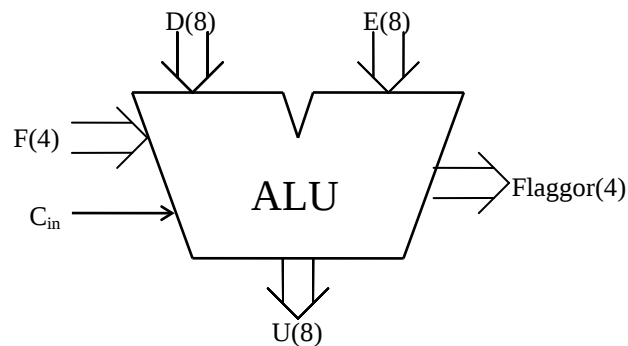
BMI Adr Ord1: 5BH(Opkod) Ord2: Offset

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC→T	OE _{PC} , LD _{MA} , LD _T .
Q ₆	M+T+1→R, PC+1→PC	MR, f ₃ , f ₁ , g ₀ , LD _R , IncPC.
Q ₇	If N=1: R→PC; else: (Ingenting)	OE _R , LD _{PC} =N, (NF).

BPL Adr Ord1: 5CH(Opkod) Ord2: Offset

State	RTN-beskrivning	Styrsignaler
Q ₅	PC→MA, PC→T	OE _{PC} , LD _{MA} , LD _T .
Q ₆	M+T+1→R, PC+1→PC,	MR, f ₃ , f ₁ , g ₀ , LD _R , IncPC.
Q ₇	If N=0: R→PC, (NF); else: (Ingenting)	OE _R , LD _{PC} =N', (NF).

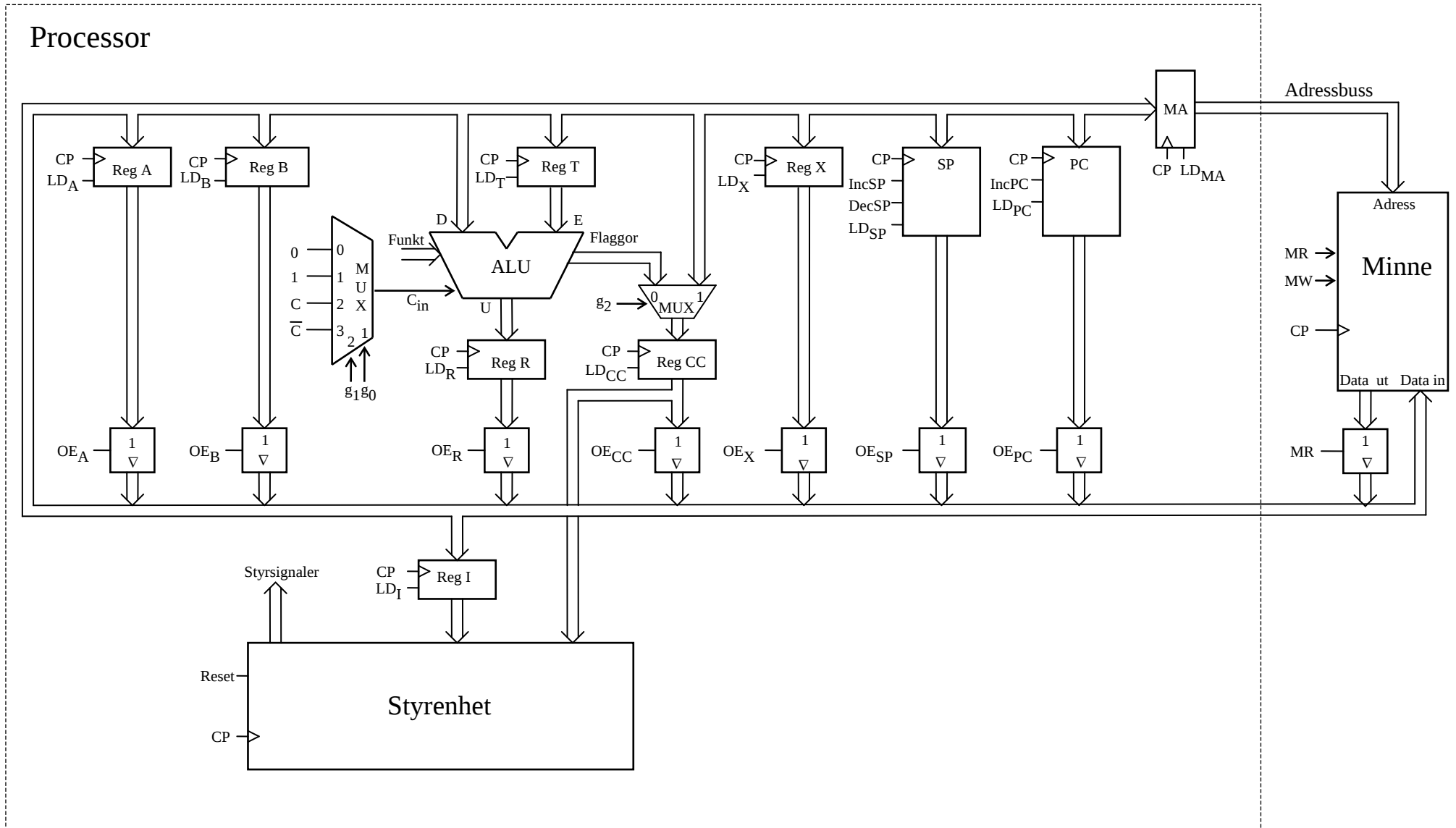
Bilaga 1

ALU:ns funktion

ALU:ns logiska och aritmetiska operationer på indata D och E definieras av ingångarna $F = (f_3, f_2, f_1, f_0)$ och C_{in} enligt tabellen nedan.

f_3 f_2 f_1 f_0	$U = f(D, E, C_{in})$
0 0 0 0	0
0 0 0 1	D
0 0 1 0	E
0 0 1 1	D'
0 1 0 0	E'
0 1 0 1	D OR E
0 1 1 0	D AND E
0 1 1 1	D XOR E
1 0 0 0	$D + C_{in}$
1 0 0 1	$D - 1 + C_{in}$
1 0 1 0	$D + E + C_{in}$
1 0 1 1	$D + D + C_{in}$
1 1 0 0	$D - E - 1 + C_{in}$
1 1 0 1	0
1 1 1 0	0
1 1 1 1	FFH

I tabellen ovan avser "+" och "-" aritmetiska operationer.
Med t ex D' menas att samtliga bitar i D inverteras.



Figur 1. Datorn FLEX.