



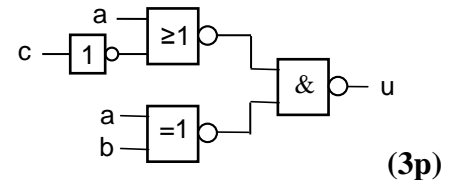
## TENTAMEN

<b>KURSNAMN</b>	<b>Digital- och datorteknik E</b>
<b>PROGRAM:</b>	<b>Elektro Åk 1/ lp 4</b>
<b>KURSBETECKNING</b>	<b>EDA216</b>
<b>EXAMINATOR</b>	<b>Lars-Eric Arebrink</b>
<b>TID FÖR TENTAMEN</b>	<b>2011-05-26 kl 8.30 – 12.30</b>
<b>HJÄLPMEDEL</b>	<b>Av institutionen utgiven ”Instruktionslista för FLEX-processorn” (INS1)  Tabellverk eller miniräknare får ej användas.</b>
<b>ANSV LÄRARE:</b> <b>Besöker tentamen</b>	<b>Lars-Eric Arebrink, tel. 772 5718 vid flera tillfällen</b>
<b>ANSLAG AV RESULTAT</b>	<b>När rättningen är färdig anslås resultatet med anonyma koder och tid för granskning på kursens hemsida.</b>
<b>ÖVRIG INFORM.</b>  <b>BETYGSGRÄNSER.</b>  <b>SLUTBETYG</b>	<b>Tentamen omfattar totalt 60 poäng. Onödigt komplicerade lösningar kan ge poängavdrag. Svar på uppgifter skall motiveras. Betyg 3: 24 poäng Betyg 4: 36 poäng Betyg 5: 48 poäng För slutbetyg 3, 4 eller 5 på kursen fordras betyg 3, 4 eller 5 på tentamen och godkända laborationer.</b>

1. I uppgift a-h nedan används 6-bitars tal X, Y, S och D.  $X = 100101$  och  $Y = 011001$ .

- a) Vilket talområde måste X, Y, S och D tillhöra om de tolkas som tal med tecken? (1p)
- b) Vilket talområde måste X, Y, S och D tillhöra om de tolkas som tal utan tecken? (1p)
- c) Visa med penna och papper hur räkneoperationen  $S = X + Y$  utförs i en 6-bitars ALU. (1p)
- d) Vilka värden får flaggbitarna N, Z, V och C vid räkneoperationen i c)? (1p)
- e) Visa med penna och papper hur räkneoperationen  $D = X - Y$  utförs i en 6-bitars ALU. (1p)
- f) Vilka värden får flaggbitarna N, Z, V och C vid räkneoperationen i e)? (1p)
- g) Tolka bitmönstren X, Y, S och D som tal *utan* tecken och ange deras decimala motsvarighet. Vilken eller vilka flaggbitar visar om resultatet är korrekt vid tal utan tecken? (1p)
- h) Tolka bitmönstren X, Y, S och D som tal *med* tecken och ange deras decimala motsvarighet. Vilken eller vilka flaggbitar anger om resultatet är korrekt vid tal med tecken? (1p)
- i) Antag att  $N_{\max}$  är det största positiva tal som kan representeras som ett 32-bitars flyttal enligt flyttalsstandarden IEEE 754-1985 (dvs 23 bitar av mantissan). Vilket decimala värde har  $N_{\max}$  approximativt. (2p)
- j) Genomför subtraktionen  $287_{10} - 859_{10}$  med 10-komplementaritmetik. Hur många decimala sifferpositioner behövs? Hur skall man tolka resultatet? (2p)

k) Ge ett "minimalt" boolesk uttryck för u i grindnätet till höger.



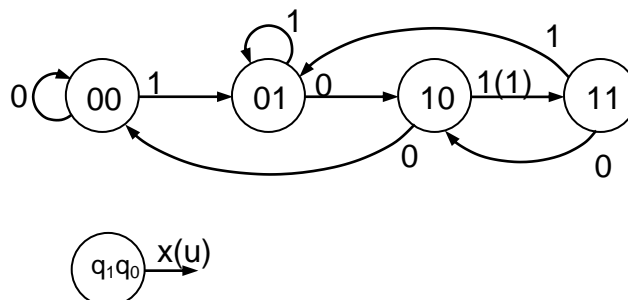
2. En boolesk funktion  $f(a,b,c,d)$  har karnaughdiagrammet till höger.

Realisera funktionen med så få grindar som möjligt. NAND-grindar med valfritt antal ingångar, XOR-grindar och NOT-grindar får användas. Endast insignalerna a, b, c och d finns tillgängliga.

		cd			
		00	01	11	10
ab	00	0	0	1	1
	01	0	1	1	1
	11	1	1	1	0
	10	1	1	0	0

(4p)

3. Gör en minimal realisering (minimant antal grindar) av ett synkront sekvensnät vars tillståndsgraf visas nedan.



(Ej utsatta utsignaler = 0)

Du får använda JK-vippor som triggar på positiv flank, NOR-grindar med valfritt antal ingångar, XOR-grindar och NOT-grindar. Sekvensnätet förutsätts starta från tillståndet där båda q-värden är "0". Inga åtgärder för att försätta sekvensnätet i detta starttillstånd behöver vidtagas. (6p)

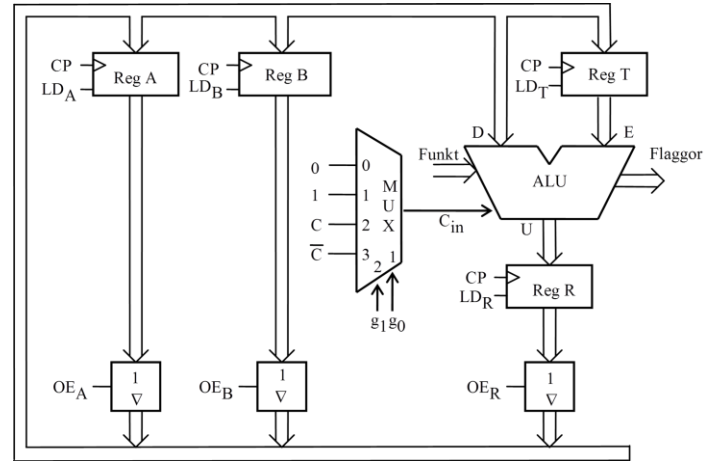
4. Ge RTN-beskrivning och styrsignaler för de tillstånd krävs för att utföra operationen enligt nedanstående RTN-beskrivning:

RTN-beskrivning:  $3 \cdot (B + 1) - 5 \cdot (A + 1) \rightarrow B$   
(Aritmetisk multiplikation avses)

Register A får inte ändras. Bortse från risken för overflow. Använd så få tillstånd som möjligt. Förutsätt att register A och B från början innehåller de data som skall behandlas enligt uttrycket ovan och att innehållena i register R och T är okända.

Använd den enkla datavägen till höger och ge ditt svar i tabellform.

Samtliga funktioner ALU:n kan utföra framgår av bilaga 1.



(5p)

5. Figur 1 i bilaga 3 visar hur datorn FLEX är uppbyggd. Bilaga 1 visar hur ALU'ns funktion väljs med styrsignalerna  $f_3 - f_0$  och signalen  $C_{in}$ .

I tabellen nedan visas styrsignalerna i EXECUTE-sekvensen för en av FLEX-processorns instruktioner.

State nr	S-term	RTN-beskrivning	Styrsignaler (=1)
Q5	$Q_5 \cdot I_{xx}$		$OE_X, LD_{MA}$
Q6	$Q_6 \cdot I_{xx}$		$MR, LD_T$
Q7	$Q_7 \cdot I_{xx}$		$OE_B, f_3, f_2, g_0, LD_{CC}, NF$

NF i tabellens sista rad anger att nästa tillstånd (state) skall vara det första i FETCH-sekvensen.

a) Ge RTN-beskrivningen för tillstånden Q5-Q7. (1p)

b) Förklara vad instruktionen med EXECUTE-sekvensen ovan utför i varje klockcykel. Skriv instruktionen med assemblerspråk för FLEX-processorn. (2p)

c) Med instruktionen nedan skall man kunna ettställa valfria bitar i ett minnesord på adressen Adr. Den skall implementeras för FLEX-processorn med hjälp av styrenheten med fast logik.

BSET Adr,#mask RTN:  $M(Adr) OR mask \rightarrow M(Adr), Flags \rightarrow CC$

Samtliga funktioner ALU:n kan utföra framgår av bilaga 1. FLEX-datorn visas i bilaga 3. Gör en tabell liknande tabellen ovan för den efterfrågade EXECUTE-sekvensen. Använd operationskoden  $FA_{16}$ .

OPKOD
Adr
mask

(4p)

6. Besvara kortfattat följande frågor rörande FLEX-processorn.

- a) Varför får man inte hoppa till subrutiner med JMP- eller BRA-instruktioner? **(2p)**
- b) I hoppvillkoren för villkorliga hopp för tal med tecken används inte N-flaggan ensam, utan den kombineras alltid med V-flaggan. Vad är skälet till detta? **(2p)**
- c) Instruktionerna BHI Adr och BGT Adr innebär bägge att hopp skall utföras om villkoret ">" är uppfyllt. Förklara skillnaden mellan instruktionerna och vilken praktisk betydelse den har. **(2p)**
- d) Översätt programavsnittet nedan till maskinkod. Det skall framgå hur "offset" för branchinstruktionerna beräknas.

```

                ORG    $20
                LDX    #$40
                LDAA   #-10
                LDAB   #5
LOOP1          INCA
                DECB
                BNE    LOOP1
LOOP2          INCB
                INCA
                BNE    LOOP2
                STAB   B,X
                EORB   #$0F
                STAB   1,-X

```

**(2p)**

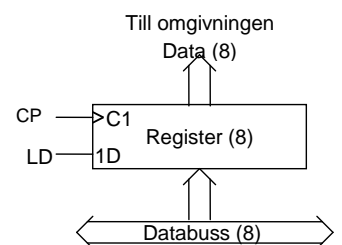
e) Vilka värden och var skriver de två STAB-instruktionerna i d) ovan i minnet.? **(1p)**

f) Hur många klockpulser krävs för att köra hela programavsnittet i d). **(3p)**

7. FLEX-datorn som visas i bilaga 3 skall kompletteras med en utport på adressen  $FE_{16}$ . Principen för utporten visas i figuren till höger. Det skall inte finnas några andra portar i datorn.

a) Konstruera ett grindnät som bildar signalen LD. Standardgrindar med valfritt antal ingångar får användas.

b) I programmet som använder utporten vill man ha möjlighet att läsa det värde som finns i utportens register, dvs det senast utmatade värdet. Visa hur man enklast kan lösa detta problem.

**(2p + 2p)**

8. Skriv en subrutin CNT2 i assemblerspråk för FLEX-processorn som söker igenom en tabell med 8-bitars tal med tecken. Subrutinen skall räkna hur många talvärden som tillhör de två intervallen  $[-75,75]$  och  $[50,100]$ . Antalet tal i det första intervallet skall returneras i A-registret och antalet tal i det andra i B-registret.

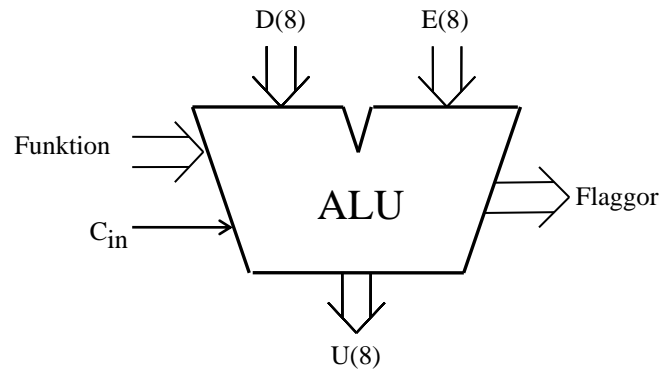
Före anrop av subrutinen placerar huvudprogrammet adressen till tabellen i X-registret. Tabellen avslutas med ett dataord med talvärdet  $10000000_2$ . X-registret skall vara oförändrat vid återhopp.

För full poäng på uppgiften skall programmet vara korrekt radkommenterat.

**(7p)**

## Bilaga 1

## ALU:ns funktion



ALU:ns **logik-** och **aritmetikoperationer** på indata **D** och **E** definieras av ingångarna **Funktion (F)** och **C<sub>in</sub>** enligt tabellen nedan. **F = (f<sub>3</sub>, f<sub>2</sub>, f<sub>1</sub>, f<sub>0</sub>)**.

I kolumnen Operation förklaras hur operationen utförs.

f <sub>3</sub> f <sub>2</sub> f <sub>1</sub> f <sub>0</sub>	U = f(D,E,C <sub>in</sub> )	
	Operation	Resultat
0 0 0 0	bitvis nollställning	0
0 0 0 1		D
0 0 1 0		E
0 0 1 1	bitvis invertering	D <sub>1k</sub>
0 1 0 0	bitvis invertering	E <sub>1k</sub>
0 1 0 1	bitvis OR	D OR E
0 1 1 0	bitvis AND	D AND E
0 1 1 1	bitvis XOR	D XOR E
1 0 0 0	D + 0 + C <sub>in</sub>	D + C <sub>in</sub>
1 0 0 1	D + FFH + C <sub>in</sub>	D - 1 + C <sub>in</sub>
1 0 1 0		D + E + C <sub>in</sub>
1 0 1 1	D + D + C <sub>in</sub>	2D + C <sub>in</sub>
1 1 0 0	D + E <sub>1k</sub> + C <sub>in</sub>	D - E - 1 + C <sub>in</sub>
1 1 0 1	bitvis nollställning	0
1 1 1 0	bitvis nollställning	0
1 1 1 1	bitvis ettställning	FFH

**Carryflaggan (C)** innehåller minnessiffran ut (carry-out) från den mest signifikanta bitpositionen (längst till vänster) om en aritmetisk operation utförs av ALU:n.

Vid **subtraktion** gäller för denna ALU att **C = 1 om lånesiffra (borrow) uppstår och C = 0 om lånesiffra inte uppstår**.

Carryflaggans värde är 0 vid andra operationer än aritmetiska.

**Overflowflaggan (V)** visar om en aritmetisk operation ger "overflow" enligt reglerna för 2-komplementaritmetik.

V-flaggans värde är 0 vid andra operationer än aritmetiska.

**Zeroflaggan (Z)** visar om en ALU-operation ger värdet noll som resultat på U-utgången.

**Signflaggan (N)** är identisk med den mest signifikanta biten (teckenbiten) av utsignalen U från ALU:n.

**Half-carryflaggan (H)** är minnessiffran (carry) mellan de fyra minst signifikanta och de fyra mest signifikanta bitarna i ALU:n.

H-flaggans värde är 0 vid andra operationer än aritmetiska.

I tabellen ovan avser "+" och "-" **aritmetiska operationer**. Med t ex **D<sub>1k</sub>** menas att samtliga bitar i **D** inverteras.

## Bilaga 2

Assemblerspråket för FLEX-processorn.

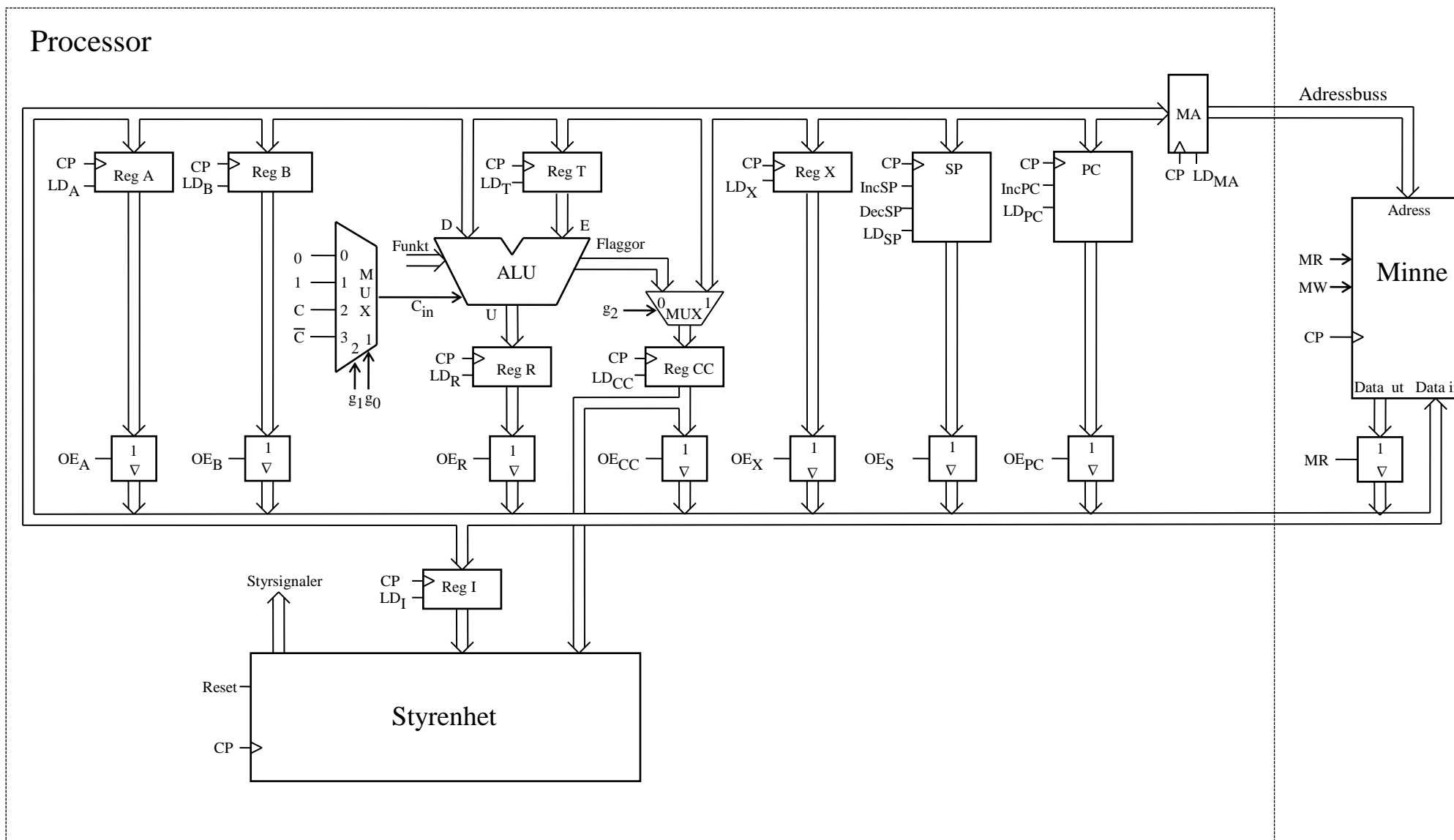
Assemblerspråket använder sig av mnemoniska beteckningar liknande dem som processorkonstruktören MOTOROLA (FREESCALE) specificerat för maskininstruktioner för mikroprocessorerna 68XX och instruktioner till assemblatorn, så som pseudoinstruktioner eller assemblatordirektiv. Pseudoinstruktionerna listas i tabell 1.

**Tabell 1**

Direktiv	Förklaring
ORG N	Placerar den efterföljande koden med början på adress N. (ORG för ORiGin = ursprung)
L RMB N	Avsätter N bytes i följd i minnet (utan att ge dem värden), så att programmet kan använda dem. Följden placeras med början på adressen L. (RMB för Reseve Memory Bytes)
L EQU N	Ger symbolen L konstantvärdet N. (EQU för EQUates = beräknas till)
L FCB N1,N2	Avsätter en byte för varje argument i följd i minnet. Respektive byte ges konstantvärdet N1, N2 etc. Följden placeras med början på adressen L. (FCB för Form Constant Byte)
L FCS "ABC"	Avsätter en byte för varje tecken i teckensträngen "ABC" i följd i minnet. Respektive byte ges ASCII-värdet för A B C, etc. Följden placeras med början på adressen L. (FCS för Form Character String)

**Tabell 2 7-bitars ASCII**

000	001	010	011	100	101	110	111	$b_6b_5b_4$ $b_3b_2b_1b_0$
NUL	DLE	SP	0	@	P	`	p	0 0 0 0
SOH	DC1	!	1	A	Q	a	q	0 0 0 1
STX	DC2	"	2	B	R	b	r	0 0 1 0
ETX	DC3	#	3	C	S	c	s	0 0 1 1
EOT	DC4	\$	4	D	T	d	t	0 1 0 0
ENQ	NAK	%	5	E	U	e	u	0 1 0 1
ACK	SYN	&	6	F	V	f	v	0 1 1 0
BEL	ETB	'	7	G	W	g	w	0 1 1 1
BS	CAN	(	8	H	X	h	x	1 0 0 0
HT	EM	)	9	I	Y	i	y	1 0 0 1
LF	SUB	*	:	J	Z	j	z	1 0 1 0
VT	ESC	+	;	K	[Ä	k	{ä	1 0 1 1
FF	FS	,	<	L	\Ö	l	ö	1 1 0 0
CR	GS	-	=	M	]Å	m	}å	1 1 0 1
S0	RS	.	>	N	^	n	~	1 1 1 0
S1	US	/	?	O	_	o	RUBOUT (DEL)	1 1 1 1



Figur 1. Datoren FLEX.