

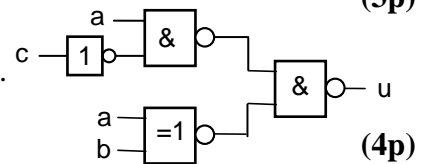
**TENTAMEN**

KURSNAMN	Digital- och datorteknik
PROGRAM:	Data-, elektro- och mekatronikingenjör Åk 1/ lp 1 och 2
KURSBETECKNING	LEU431
EXAMINATOR	Lars-Eric Arebrink
TID FÖR TENTAMEN	2012-08-24 kl 14.00 – 18.00
HJÄLPMEDEL	Av institutionen utgiven ”Instruktionslista för FLEX-processorn” (INS1) Tabellverk eller miniräknare får ej användas.
ANSV LÄRARE: Besöker tentamen	Lars-Eric Arebrink, tel. 772 5718 vid flera tillfällen.
ANSLAG AV RESULTAT	När rättningen är färdig anslås resultatet med anonyma koder och tid för granskning på kursens hemsida. Lägg din anonyma kod på minnet. Den används när resultatet anslås!
ÖVRIG INFORM.	Tentamen omfattar totalt 60 poäng. Onödigt komplicerade lösningar kan ge poängavdrag. Svar på uppgifter skall motiveras.
BETYGSGRÄNSER.	Betyg 3: 24 poäng Betyg 4: 36 poäng Betyg 5: 48 poäng
SLUTBETYG	För slutbetyg 3, 4 eller 5 på kursen fordras betyg 3, 4 eller 5 på tentamen och godkända laborationer.

1. I uppgift a-h nedan används 8-bitars tal X, Y, S och D. $X = 11010111$ och $Y = 00101001$.

- a) Vilket talområde måste X, Y, S och D tillhöra om de tolkas som tal *utan* tecken? (1p)
- b) Vilket talområde måste X, Y, S och D tillhöra om de tolkas som tal *med* tecken? (1p)
- c) Visa med penna och papper hur räkneoperationen $S = X + Y$ utförs i en 8-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 8-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) Det positiva talet N skall packas som ett 32-bitars flyttal med full upplösning enligt flyttalsstandarden IEEE 754-1985 (dvs 23 bitar av mantissan). Visa approximativt vilket talområde N måste tillhöra för att detta skall vara möjligt. (3p)

j) Ge ett "minimalt" boolesk PS-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	1	1	0	0
	11	0	1	1	0
	10	0	1	1	0

(4p)

3. Realisera en JK-vippa med hjälp av en T-vippa och standardgrindar.

(5p)

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

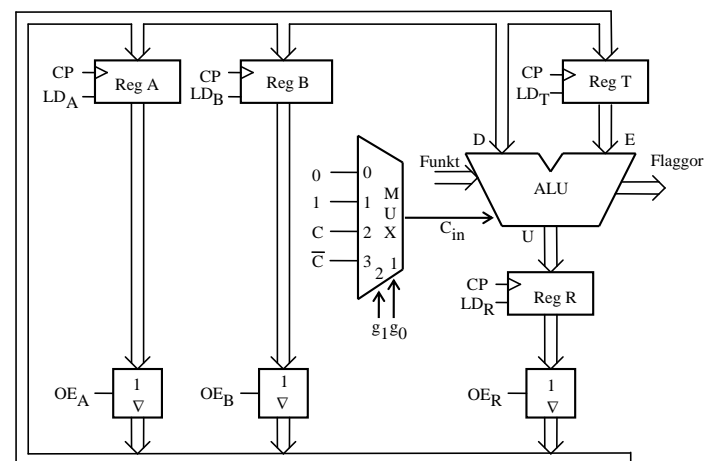
RTN-beskrivning: $7 \cdot A + 3 \cdot B \rightarrow A$
(Aritmetisk multiplikation avses)

Register B 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} . OE_{PC}

I tabellen nedan visas styrsignalerna för ett antal tillstånd i FLEX-processorns styrenhet.

State	S-term	RTN-beskrivning	Styrsignaler (= 1)
Q_y	$Q_y I_{xx}$		f_3, f_2, f_1, f_0, LD_R
Q_{y+1}	$Q_{y+1} \cdot I_{xx}$		OE_R, LD_{MA}
Q_{y+2}	$Q_{y+2} \cdot I_{xx}$		MR, LD_{PC}, NF

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

- a) Ge RTN-beskrivningen för tillstånden och förklara vad tillståndssekvensen utför. (2p)
- b) Med instruktionen nedan skall man kunna invertera valfria bitar i ett minnesord. De bitpositioner som skall inverteras skall vara ettställda i instruktionens andra ord, mask. Register A, B, X eller SP får inte påverkas av instruktionen som skall implementeras för FLEX-processorn med hjälp av styrenheten med fast logik.

BINV Adr,#mask RTN: $M(\text{Adr}) \text{ XOR mask} \rightarrow M(\text{Adr}), \text{Flags} \rightarrow \text{CC}$

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 FC_{16} .

OPKOD
mask
Adr

(4p)

6. Besvara kortfattat följande frågor rörande FLEX-processorn.

- a) Förklara kortfattat vad som händer när processorn utför FETCH-fasen. (2p)
- b) Förklara kortfattat hur stacken fungerar. (2p)
- c) Det finns två principer för att ansluta inportar och utportar till processorns bussar, separatadresserad resp. minnesorienterad in- och utmatning. Vilken av dessa används i FLEX-processorn? Vilka är fördelarna med denna metod? (2p)

För vilka värden på dataordet W ($0 \leq W \leq 255$) utförs hoppen i d) och e)?

- d) LDAA # $\$3F$
 ADDA # W
 BLO Hopp (2p)
- e) LDAA # W
 NEGA
 ADDA # $\$55$
 BLT Hopp (4p)

6. forts.

- f) Översätt programavsnittet till höger till maskinkod. Det skall framgå hur "offset" för branchinstruktionerna beräknas.

(3p)

- g) Hur många klockpulser krävs för att köra hela programavsnittet i f)?

(3p)

	ORG	\$40
	LDAA	#20
	LDX	#TAB
LOOP	LDAB	1,X+
	STAB	\$FE
	DECA	
	BPL	LOOP
	LDAB	12
	ANDB	#\$0F
	BRA	NEXT
TAB	RMB	\$14
NEXT	NOP	

7. I simulatören för FLEX-datorn kan man ansluta strömbrytarmodulen DIPSWITCH till en inport och sifferindikatorn HEXDISPLAY till en utport. På DIPSWITCH kan man ställa in ett 8-bitars datavärde och på HEXDISPLAY kan man visa ett 8-bitars dataord som ett hexadecimalt tal med två siffror.

Skriv ett program som läser av DIPSWITCH och visar ett värde på HEXDISPLAY. Det visade värdet skall bero av det avlästa värdet enligt tabellen nedan. Programmet skall utformas som en evighets-slinga. DIPSWITCH skall läsas av en gång i början på varje varv och motsvarande värde enligt tabellen skall sedan visas på HEXDISPLAY.

Om inget av villkoren i tabellen är uppfyllt skall värdet EE_{16} visas på HEXDISPLAY.

Rita en flödesplan som beskriver programmet och skriv det i assemblerspråk för FLEX-processorn. Startadressen för programmet skall vara 50_{16} .

DIPSWITCH är ansluten till inport FD_{16} och HEXDISPLAY till utport FE_{16} .

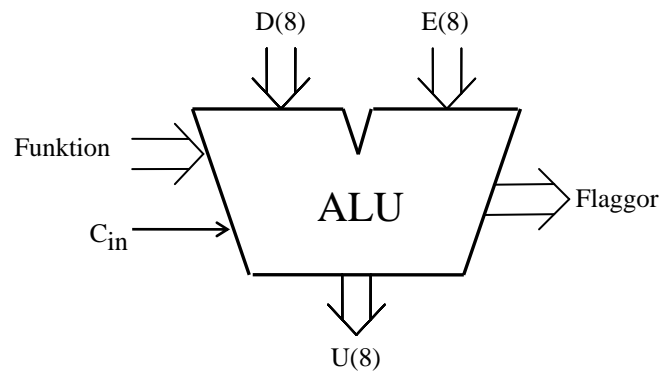
För full poäng på uppgiften skall programmet vara korrekt radkommenterat.

(2p+5p)

Villkor	DIPSWITCH								HEXDISPLAY
	b7	b6	b5	b4	b3	b2	b1	b0	
1	1	1	?	0	?	0	?	1	\$10
2	?	?	?	1	?	?	1	?	\$23
3	0	1	0	0	1	1	1	0	\$47

Bilaga 1

ALU:ns funktion



ALU:ns **logik-** och **aritmetikoperationer** på indata **D** och **E** definieras av ingångarna **Funktion (F)** och **C_{in}** enligt tabellen nedan. **F = (f₃, f₂, f₁, f₀)**.

I kolumnen Operation förklaras hur operationen utförs.

f ₃ f ₂ f ₁ f ₀	U = f(D,E,C _{in})	
	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 _{1k}
0 1 0 0	bitvis invertering	E _{1k}
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 _{in}	D + C _{in}
1 0 0 1	D + FFH + C _{in}	D - 1 + C _{in}
1 0 1 0		D + E + C _{in}
1 0 1 1	D + D + C _{in}	2D + C _{in}
1 1 0 0	D + E _{1k} + C _{in}	D - E - 1 + C _{in}
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_{1k}** menas att samtliga bitar i **D** inverteras.

Bilaga 2

Assemblerspråket för FLEX-processorn.

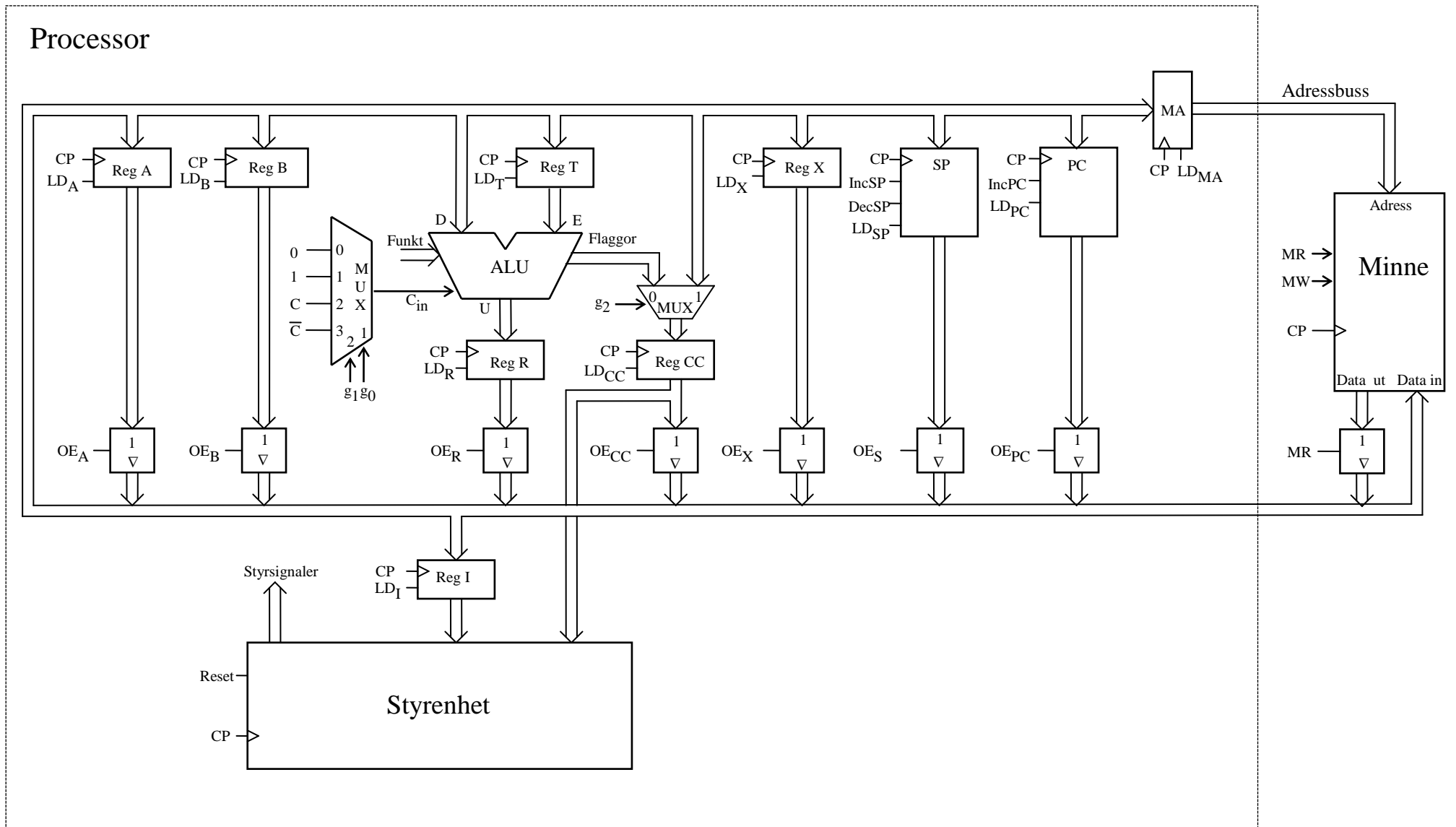
Assemblerspråket använder sig av mnemoniska beteckningar liknande dem som processorkonstruktören MOTOROLA specificerat för maskininstruktioner för mikroprocessorerna 68XX och instruktioner till assemblatorn, så som pseudoinstruktioner eller assemblatordirektiv. Pseudoinstruktionerna listas i tabell 1.

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

0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	$b_6b_5b_4$ $b_3b_2b_1b_0$
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. Datorn FLEX.