



Tentamen med lösningsförslag

EDA451 Digital- och datorteknik, D

EDA215 Digital- och datorteknik, Z

DIT790 Digital- och datorteknik, GU

Tisdag 14 April 2009, kl. 8.30 - 12.30, VV-salar

Examinatorer

Roger Johansson, tel. 772 5729

Rolf Snedsböl, tel 772 1665

Kontaktperson under tentamen

Rolf Snedsböl

Tillåtna hjälpmedel

Häften

Instruktionslista för FLEX

Instruktionslista för CPU12

I dessa får rättelser och understrykningar vara införda, inget annat.

Tabellverk och miniräknare får ej användas!

Lösningar

anslås via kursens hemsida.

Granskning

Tid och plats anges på kursens hemsida.

Allmänt

Siffror inom parentes anger full poäng på uppgiften. **För full poäng krävs att:**

- redovisningen av svar och lösningar är läslig och tydlig.
- ett lösningsblad får endast innehålla redovisningsdelar som hör ihop med en uppgift.
- lösningen ej är onödigt komplicerad.
- du har motiverat dina val och ställningstaganden
- redovisningen av hårdvarukonstruktioner innehåller funktionsbeskrivning, lösning och realisering.
- redovisningen av mjukvarukonstruktioner i assembler är dokumenterad med radkommentarer, om inget annat anges i uppgiften.

Betygsättning

För godkänt slutbetyg på kursen fordras att både tentamen och laborationer är godkända. Tentamen ger slutbetyget:

$20p \leq \text{betyg 3} < 30p \leq \text{betyg 4} < 40p \leq \text{betyg 5}$

Uppgift 1

Talomvandling, tvåkomplementsaritmetik

I uppgift a-f nedan används 5-bitars tal. $X = 10101$ och $Y = 01011$.

- Visa med penna och papper hur räkneoperationen $R = X+Y$ utförs i en dator (i en ALU). **(1p)**
- Ange hur flaggbitarna N, Z, V, C påverkas av operationen. **(1p)**
- Visa med penna och papper hur räkneoperationen $R = X-Y$ utförs i en dator (i en ALU). **(1p)**
- Ange hur flaggbitarna N, Z, V, C påverkas av operationen. **(1p)**
- Tolka bitmönstren R, X och Y som tal *utan* tecken och ange dess decimala motsvarighet. Vilken(vilka) flaggbit(ar) anger om resultatet är korrekt då bitmönstren tolkas som tal utan tecken? **(1p)**
- Tolka bitmönstren R, X och Y som tal *med* tecken och ange dess decimala motsvarighet. Vilken(vilka) flaggbit(ar) anger om resultatet är korrekt då bitmönstren tolkas som tal utan tecken? **(1p)**
- Vad blir det trebitars Gray-kodade ordet för talet 3_{10} då Gray-kodordet 000 representerar 0_{10} . **(2p)**

Lösningsförslag:

a.

$$\begin{array}{r} X \\ +Y \\ \hline =R \end{array} \quad \begin{array}{r} 11111 \\ 10101 \\ + 01011 \\ \hline = 00000 \end{array}$$

b. $N=0, Z=1, V=0$ (ty $X_4 \neq Y_4$), $C_5=1 \Rightarrow C=1$

c. $R=X-Y$ utförs som $R=X+Y_{1k}+1$; $Y_{1komp} = 10100$.

$$\begin{array}{r} X \\ +Y_{1komp} \\ \hline =R \end{array} \quad \begin{array}{r} 101011 \\ 10101 \\ + 10100 \\ \hline = 01010 \end{array}$$

d. $N=0; Z=0; V=1$ (ty $X_4 = Y_4$ och $S_4 \neq Y_4$); $C_5=1 \Rightarrow C=0$

e. $X=21$ $Y=11$; $R=10$

$C=0$ anger att resultatet är korrekt vid tal utan tecken. (Kontroll: $21-11=10$; Ok!)

f. $X = -11$; $Y = 11$; $R = 10$

$V = 1$ anger fel vid tal med tecken. (Kontroll: $-3-(11) = 10$; FEL!)

g. $3_{10} = 010_{\text{GRAY}}$.

Uppgift 2

- a. Ange funktions- och excitationstabell för en D-vippa (1p)

En boolesk funktion har fyra variabler x, y, z och w . Variablerna bildar ett binärtal $(xyzw)_2 \in [0, 15]_{10}$. Funktionen har sina mintermer när $xyzw$ är 0, 1, 4, 8, 9, 11 och 15. Detta kan också uttryckas som: $f(x, y, z, w) = \sum m(0, 1, 4, 8, 9, 11, 15)$

- b. Ange den disjunktiva normalformen för f (1p)
 c. Minimera f med hjälp av Karnaughdiagram (3p)
 d. Realisera f med NAND-logik, (endast 3-ingångars NAND-grindar får användas) förutsatt att inverser (x', y', z', w') finns tillgängliga. (3p)

Lösningsförslag

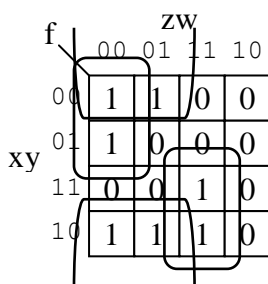
a.

D-vippan			
Funktionstabell		Excitationstabell	
D	Q ⁺	Q	Q ⁺
0	0	0	0
1	1	0	1
		1	0
		1	1

b.

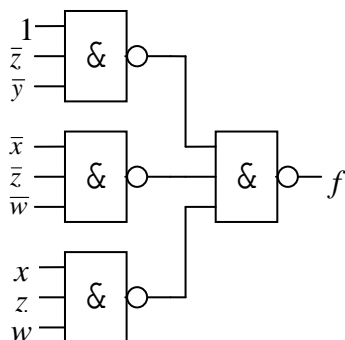
$$f = \bar{x} \bar{y} \bar{z} \bar{w} + \bar{x} \bar{y} \bar{z} w + \bar{x} y \bar{z} \bar{w} + x \bar{y} \bar{z} \bar{w} + x \bar{y} \bar{z} w + x \bar{y} z w + x y z w$$

c.



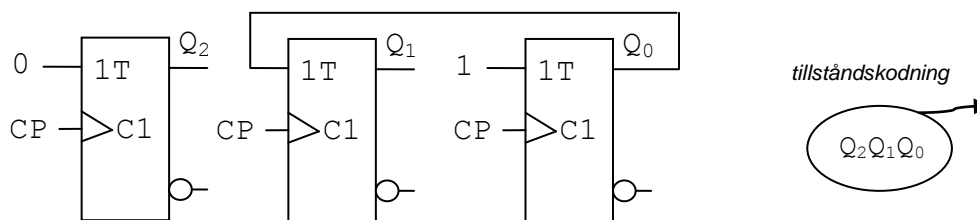
$$f = \bar{y} \bar{z} + \bar{x} \bar{z} \bar{w} + xzw$$

d.



Uppgift 3

I följande figur visas ett sekvensnät med T-vippor.



a. Analysera kopplingen, dvs. visa tillståndstabell och tillståndsgraf. **(4p)**

b. Antag att begynnelsestillståndet alltid är 000 och beskriv hur kopplingen då kan förenklas. **(2p)**

Lösningsförslag:

a. För vippornas T-ingångar gäller

$$T_2 = 0$$

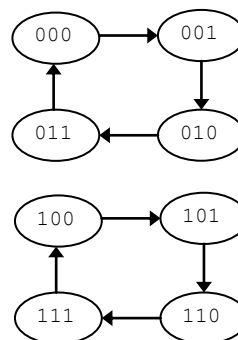
$$T_1 = Q_0$$

$$T_0 = 1$$

Tillståndstabellen blir

Q_2	Q_1	Q_0	T_2	T_1	T_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	0	0	1	0	1	1
0	1	1	0	1	1	0	0	0
1	0	0	0	0	1	1	0	1
1	0	1	0	1	1	1	1	0
1	1	0	0	0	1	1	1	1
1	1	1	0	1	1	1	0	0

och tillståndsgraf



b.

Då begynnelsestillståndet alltid är 000 kan bara tillstånden 000, 001, 010 och 011 antas, vi kan därför ta bort vippan T_2 .

Uppgift 4

- a. I tabellen nedan visas RTN-beskrivningen för EXECUTE-sekvensen hos en **instruktion** för FLEX-processorn. (NF i tabellens sista rad anger att nästa tillstånd (state) skall vara det första i en ny FETCH-sekvens). Rita en tabell där du anger tillstånden nr (5,6,7) och dess styrsignaler. Endast styrsignaler = 1 skall anges. Du kan utelämna RTN-beskrivningen i din tabell

Tillstånd	RTN-beskrivning	Styrsignaler=1
5	A→T, 00→R,	
6	R-T→R, Flaggor→CC	
7	R→A (NF)	

Förklara också i ord vad instruktionen ovan utför i respektive klockcykel. Ange slutligen instruktionen med assemblerspråk för FLEX-processorn (Ex: LDAA Adr). **(3p)**

- b. Ange styrsignalsekvensen för EXECUTE-fasen för FLEX-instruktionen ROLA. (Studera instruktionslistan). Ge ditt svar i tabellform liknande den ovan. Såväl RTN-beskrivning som styrsignaler ska anges. **(3p)**

- c. Man vill införa en ny instruktion i FLEX-datorn:

DBEQ Decrement register and branch on equal

Instruktion: DBEQ A,Adr

RTN: A-1→A; if A = 0: PC+Offset → PC

Flaggor: N: Ettställs om resultatets teckenbit (bit 7) får värdet 1.
Z: Ettställs om samtliga åtta bitar i resultatet blir noll.
V: Ettställs om 2-komplementoverflow uppstår.
C: Ettställs om borrow uppstår.

Beskrivning: Minskar innehållet i register A med 1. Testar därefter innehållet i register A. Om A=0 utförs ett hopp till adressen Adr = PC+Offset. Offset räknas från adressen efter branchinstruktionen, dvs vid uträkningen av hoppadressen pekar PC på operationskoden direkt efter branchinstruktionen i minnet. Om A≠0 utförs inget hopp. Nästa instruktion blir i så fall den direkt efter branchinstruktionen i minnet

Instruktionsformat och OP-kod:

F1	offset
----	--------

Allokering i mikrominne, adresser 4C0-4CF.

Ledning: Instruktionen sätts samman av de båda instruktionerna DECA, BEQ <rel8>

Visa hur instruktionens exekveringsfas kan implementeras hos en FLEX med mikroprogrammerad styrenhet. Svara ”i tabellform” exempelvis genom att kopiera följande uppställning: **(6p)**

Adress	Transfer-villkor G_K	Transfer-adress	Aktiva styrsignaler	RTN-beskrivning

Lösningsförslag:

a.

Tillstånd	RTN-beskrivning	Styrsignaler=1
5	A→T, 00 ₁₆ →R,	OE _A , LD _T , LD _R .
6	R-T→R, Flaggor→CC	OE _R , f ₃ , f ₂ , g ₀ , LD _R , LD _{CC} .
7	R→A	OE _R , LD _A , NF.

(5) Bilda operationen 0 – (A)

(6) Utför operationen och sätt flaggor i CC

(7) Återför resultatet till ack A

Instruktionen är NEGA

b)

ROLA

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

c.

Mikroprogram för FLEX				
Instruktion (mnemonic)			DBEQ A, <rel8>	
Instruktionsformat			Allokering i mikrominne	4C0 - 4CF
F1	rel8			
Adress	Transfer-villkor G _K	Transfer-adress	Aktiva styrsignaler	RTN-beskrivning
F1	G ₀ '=1	4B0	LD _{CC} , LD _R , OE _A , f ₃ , f ₀	A-1→R, Flaggor→CC
4B0	G ₀ =0		LD _A , OE _R	R→A
4B1	G ₀ =0		LD _{MA} , OE _{PC} , LD _T	PC→MA, PC→T
4B2	G ₃ =Z	4B4	MR, f ₃ , f ₁ , g ₀ , LD _R	M+T+1→R
4B3	1	108		

Uppgift 5

- a. Översätt följande assemblerprogram för FLEX till maskinkod. Rita en figur av minnet där det klart framgår hur maskinkod och assemblerkod hör ihop och vilka minnesadresser maskinkoden är placerad på. **(6p)**

```

PORT    EQU    $FE
Langd   EQU    4

        ORG    $20
Start   LDX    #Tabell
        LDAA   Langd
Loop    LDAB   1,X+
        STAB   PORT
        DECA
        BNE   Loop
Stop    JMP    Stop

Tabell  FCB    $10,%10,10,0
    
```

Ange hexadecimalt de talintervall (i register A) för vilka hoppen utförs i följande instruktionssekvenser:

- b. CMPA #\$80 **(2p)**
 BLO ...

- c. CMPA #\$80 **(2p)**
 BLT ...

Lösningsförslag:

a.

Adr	Kod			
			ORG	\$20
20	11	Start	LDX	#Tabell
21	2C			
22	0B		LDAA	Langd
23	04			
24	7C		LDAB	1,X+
25	14		STAB	PORT
26	FE			
27	44		DECA	
28	5E		BNE	Loop
29	FA			
2A	59	Stop	JMP	Stop
2B	2A			
2C	10	Tabell	FCB	\$10,%11,10,0
2D	02			
2E	0A			
2F	00			

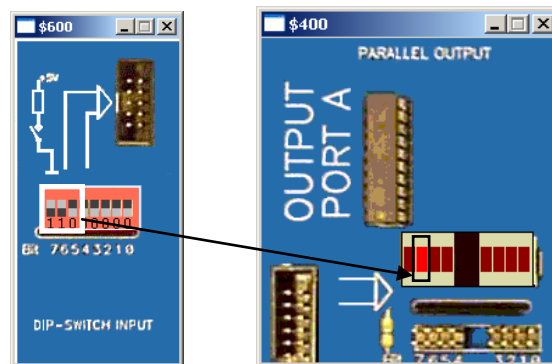
- b. \$00-\$7F
 c. Hoppas aldrig (\$80 är det minsta talet)

Uppgift 6

Vid simulatorpassen och i labbet använde du strömbrytarna (ML4 INPUT) och ljusdiodindikatorn (ML4 OUTPUT). Du skall skriva ett assemblerprogram för CPU12 som om och om igen läser strömbrytarna och skriver till ljusindikatorn.

Programmet skall tända exakt en diod hos indikatorn. Bitpositionen för den diod som ska tändas avläses från de tre mest signifikanta bitarna (b_7 b_6 b_5). Dessa bitar tolkas som ett binärt tal (000-111). Det binära talet 000 anger att dioden i den minst signifikanta positionen, b_0 (längst till höger) ska tändas, det binära talet 110 (se figurer) innebär att dioden näst längst till vänster (b_6) ska tändas, osv. för alla 8 dioderna.

Programmets startadress skall vara \$1000



Följande definitioner gäller:

Inport	EQU	\$600	; Adress för inport (Strömbrytare)
Utport	EQU	\$400	; Adress för utport (Sifferindikator)

(6p)

Lösningsförslag:

```
; Lösningsförslag:
      ORG      $1000
Start  LDAB   Inport      ; Läs inporten
; skifta till rätt position
      RORB
      RORB
      RORB
      RORB
      RORB
      RORB
      ANDB   #7
; Skapa Bitmask
      LDAA   #1
Testa:  TSTB
      BEQ   Skriv
      ROLA
      DECB
      BRA   Testa
Skriv:
      STAA  $400
      BRA  Start
```