



Tentamen med lösningsförslag

EDA441 Digital- och datorteknik, D

EDA215 Digital- och datorteknik, Z

DIT790 Digital- och datorteknik, GU

Måndag 15 December 2008, kl. 8.30 - 12.30

Examinatorer

Roger Johansson, tel. 772 5729

Rolf Snedsböl, tel 772 1665

Kontaktperson under tentamen

Som ovan

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 fullständigt dokumenterad, dvs. är redovisad både i strukturform (flödesplan eller pseudospråk) och med kommenterat program i assemblerspråk, 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-d nedan används 5-bitars tal. $X = 11101$ och $Y = 01000$.

- Visa med penna och papper hur räkneoperationen $R = X - Y$ utförs i en dator (i en ALU). **(1p)**
- Ange sedan flaggbitarna N, Z, V, C . **(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 vid 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 vid tal med tecken? **(1p)**
- Graykod och paritet.* Du har en 3-bitars Graykod $[b_3, b_1]$ och en paritetsbit (b_0). Vad blir det fyra bitars kodordet om du vill representera 5_{10} ? Använd udda paritet. Låt Graykod-ordet 000 representera decimalt 0. **(3p)**

Lösningsförslag:

- $R = X - Y$ utförs som $R = X + Y_{1k} + 1$; $Y_{1komp} = 10111$.
- $N=1; Z=0; V=0; C_5=1 \Rightarrow C=0$
- $X=29; Y=8; R=21$ (Kontroll: $29-8=21$; verkar rimligt ty $C=0$)
 $C=1$ anger att resultatet är fel vid tal utan tecken
- $X = -3; Y = 8; R = -11$ (Kontroll: $-3-(8) = -11$; verkar rimligt ty $V=0$)
 $V=1$ anger fel vid tal med tecken.
- $5_{10} = 111_{GRAY}$; Kodordet blir 1110.

X	111111
+ Y_{1komp}	+ 10111
$= R$	$= 10101$

Uppgift 2

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

Vi har två booleska funktioner:

$$g(x,y,z) = x y' z' + x' y' z + xyz + x' y z'$$

$$f(x,y,z) = (x \oplus y \oplus z)'$$

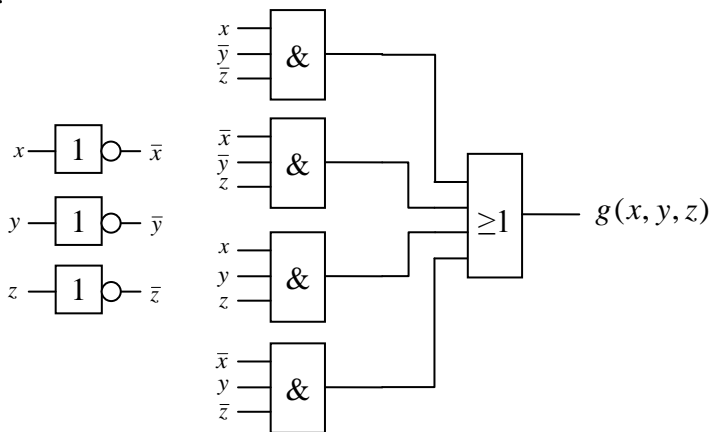
- b. Realisera $g(x,y,z)$ med AND,OR och INVERTERARE (2p)
 c. Realisera $f(x,y,z)$ med AND,OR, INVERTERARE och högst en XOR. (2p)
 d. Visa, med valfri metod, att $g = f'$ (2p)

Lösningförslag

a.

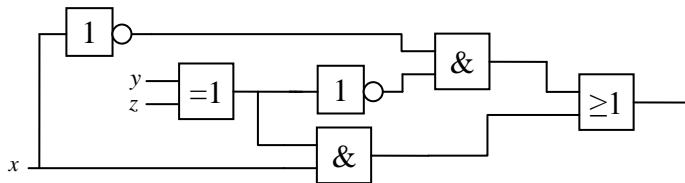
T-vippan			
Funktionsstabell		Excitationstabell	
T	Q'	Q Q'	T
0	Q	0 0	0
1	Q'	0 1	1
		1 0	1
		1 1	0

b.



c.

$$f(x,y,z) = \overline{x \oplus y \oplus z} = \overline{\bar{x}(y \oplus z) + x(y \oplus z)}$$



d.

Metod 1, algebraiskt:

$$g(x,y,z) = x\bar{y}\bar{z} + \bar{x}y\bar{z} + xyz + \bar{x}y\bar{z} = x(\bar{y}\bar{z} + yz) + \bar{x}(\bar{y}\bar{z} + yz) = x(y \oplus z) + \bar{x}(y \oplus z) = x \oplus y \oplus z = \bar{f}$$

vsv.

Metod 2, sätt upp Karnaugh-diagram (eller funktionstabell) för f och g

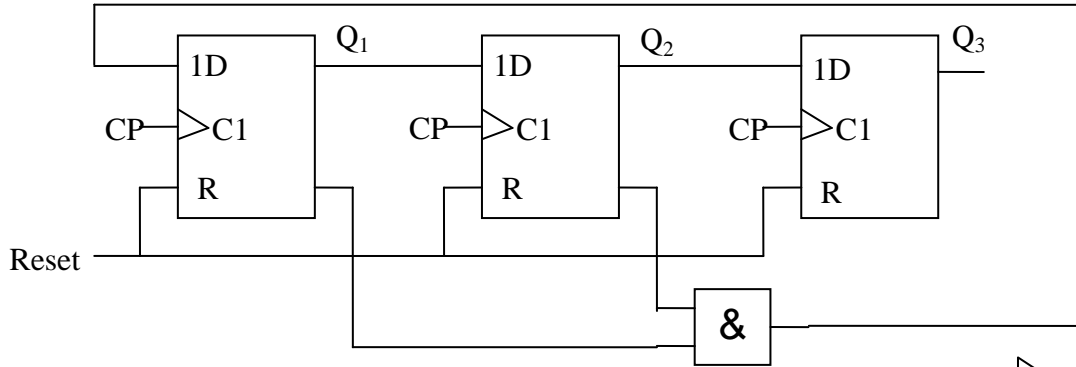
f	yz			
	00	01	11	10
0		1		1
x	1			
	1		1	

g	yz			
	00	01	11	10
0	1		1	
x	1			
		1		1

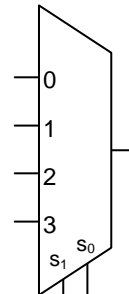
Dvs: $g = \bar{f}$, vsv.

Uppgift 3

I följande figur visas ett sekvensnät med D-vippor.



- Analysera kopplingen, dvs. Visa tillståndstabell och tillståndsgraf. **(4p)**
- Den booleska funktionen: $f(x,y,z) = x y + y' z'$ ska realiseras med hjälp av en "1 av 4 väljare" (se figur). Bestäm den lösning där x och y används som väljarsignaler. **(4p)**



Lösningsförslag:

- För vippornas D-ingångar gäller

$$D_1 = Q_1' Q_2'$$

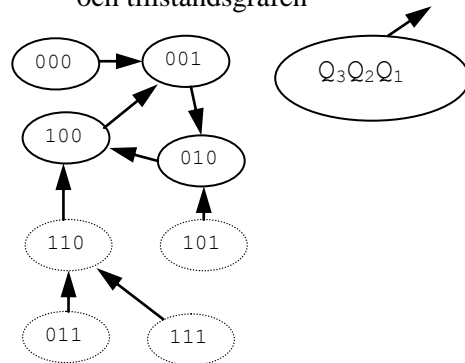
$$D_2 = Q_1$$

$$D_3 = Q_2$$

Tillståndstabellen blir

Q_3	Q_2	Q_1	D_3	D_2	D_1	Q_3^+	Q_2^+	Q_1^+
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	1	0	0	1	0	0
0	1	1	1	1	0	1	1	0
1	0	0	0	0	1	0	0	1
1	0	1	0	1	0	0	1	0
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	1	1	0

och tillståndsgrafen



-

$$f(x, y, z) = xy + \bar{y}\bar{z} = (\bar{z} + z)xy + (\bar{x} + x)\bar{y}\bar{z} = xy\bar{z} + xyz + \bar{x}\bar{y}\bar{z} + x\bar{y}\bar{z}$$

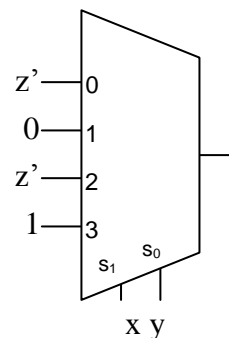
$$= \bar{x}(\bar{y}\bar{z}) + x(y\bar{z} + yz + \bar{y}\bar{z}) = \bar{y}(\bar{x}\bar{z} + x\bar{z}) + y(x\bar{z} + xz)$$

$$= \bar{z}(xy + \bar{x}\bar{y} + x\bar{y}) + z(xy)$$

Ur uttrycket kan följande funktionstabell ställas upp:

$X Y$	f
00	z'
01	0
10	z'
11	$z'+z=1$

Varför lösningen blir:



Uppgift 4

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

State	RTN-beskrivning	Styrsignaler (=1)
5	S - 1→S	
6	S→MA	
7	B→M, NF	

Förklara sedan i ord vad instruktionen ovan utför i varje klockcykel. Ange slutligen instruktionen med assemblerspråk för FLEX-processorn (Ex: LDAA Adr). **(2p)**

- b. Ange styrsignalsekvensen för EXECUTE-fasen för FLEX-instruktionen ANDCC #DATA. (Studera instruktionslistan). Ge ditt svar i tabellform liknande den ovan. **(3p)**

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

TBEQ Test register and branch on equal

Instruktion: TBEQ A,Adr

RTN: If register 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: Testar 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:

F4	offset
----	--------

Allokering i mikrominne, adresser 4C0-4CF.

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

Visa hur instruktionens exekveringsfas kan implementeras hos en FLEX med mikroprogrammerad styrenhet. Svara ”i tabellform” exempelvis genom att kopiera följande uppställning: **(5p)**

Adress	Transfer-villkor G _K	Transfer-adress	Aktiva styrsignaler	RTN-beskrivning

Lösningsförslag:

a.

State	RTN-beskrivning	Styrsignaler (=1)
5	S - 1→S	DecS.
6	S→MA	OE _S , LD _{MA} .
7	B→M, NF	OE _B , MW, NF.

- (5) Minska stackpekaren
- (6) Förbered för skrivning till stack
- (7) Skriv B till stacken; New Fetch

Instruktionen är PSHS B

b)

ANDCC #DATA

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

c.

TBEQ Test register and branch on equal

Mikroprogram för FLEX				
Instruktion (mnemonic)			TBEQ A, <rel8>	
Instruktionsformat			Allokering i mikrominne	4C0 - 4CF
F4	rel8			
Adress	Transfer-villkor G _K	Transfer-adress	Aktiva styrsignaler	RTN-beskrivning
F4	G ₀ '=1	4C0	LD _{CC} , LD _R , OE _A , f ₃ , f ₀	A→R, Flags→CC
4C0	G ₀ =0		LD _{MA} , OE _{PC} , LD _T	PC→MA, PC→T
4C1	G ₃ =Z	4C3	MR, f ₃ , f ₁ , g ₀ , LD _R IncPC	M+T+1→R PC+1→PC
4C2	1	108		
4C3	1	108	OE _R , LD _{PC}	R→PC

Uppgift 5

- a. Följande figur illustrerar några fragment minne hos en FLEX-dator. I minnet finns en ”branch”-instruktion samt dess mål indikerade. Bestäm instruktionens operand, dvs det värde som finns lagrat på adress 63. **(2p)**
- b. Ingen av ”push”-instruktionerna i en FLEX-dator i labbet fungerar. Använd andra FLEX-instruktioner och skriv en instruktionssekvens som utför PSHB. **(3p)**
- c. 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. **(5p)**

```

Konst    EQU    14
Plats    EQU    10
          ORG    $48
Start    LDX    #Tabell
          LDAB   #Konst
          LDAA   -3,X
          DECB
          STAA   Plats
          JMP    Start
Tabell   FCB    $FE,%11,11
    
```

Lösningsförslag:

- a. $OPERAND = Adress - (PC+2) = 10 - (62+2) = AC$
- b. EXG x, sp
 STAB 1, -x
 EXG x, sp
- c.

Adr	Kod			
			ORG	\$48
48	11	Start	LDX	#Tabell
49	53			
4A	10		LDAB	#Konst
4B	0E			
4C	81		LDAA	-3,X
4D	FD			
4E	45		DECB	
4F	13		STAA	Plats
50	0A			
51	59		JMP	Start
52	48			
53	FE	Tabell	FCB	\$FE,%11,11
54	03			
55	0B			

Uppgift 6

Vid simulatorpassen och i labbet använde du stömbrytarna (ML4 INPUT) och sifferindikatorn (ML4 OUTPUT).

Du skall skriva ett program för CPU12, med start på adress \$1000, som utför en fördröjning – och visar fördröjningen på en sifferindikator. Det inställda värdet på strömbrytarna anger hur lång fördröjningen är. Du har tillgång till en subrutin DELAY1s som utför en fördröjning på 1 s.

Du skall skriva en programsekvens som

- 1) läser strömbrytarna (Inport, 8 bitar, som anger antal sekunders fördröjning)
- 2) maskar fram bit $[b_3, b_0]$ av inporten (för att tillåta maximal fördröjning om 15s)
- 3) och visar detta värde $[0, F_{16}]$ på sifferindikatorn och
- 4) därefter utför fördröjningen (Hoppa till DELAY1s ett visst antal gånger) och
- 5) slutligen släcker sifferindikatorn.

Du har tillgång till en tabell med segmentkoder och följande definitioner:

Inport	EQU	qqqq	Adress för inport (Strömbrytare)
Utport	EQU	zzzz	Adress för utport (Sifferindikator)
SegCode	FCB	xx, yy, zz, etc	Tabell med segmentkoder för $[0, F]$
DELAY1s	EQU	www	Startadress för Delay-rutinen som utför 1s fördröjning

(8p)

Lösningsförslag:

```

                ORG    $1000
Start          LDX    #SegCode      ; Pekare till tabell
                LDAB  Inport        ; Läs inporten
                ANDB  #$0F          ; Maskera b7-b4
                TSTB                    ; Någon fördröjning???
                BEQ   SLUT          ; Hoppa om NEJ

                LDAA  B,X           ; Hämta kod
                STAA  Utport        ; . och visa

Loop          JSR    DELAY1s        ; .. och vänta
; (Förutsätter att DELAY inte ändrar Register A)

                DECB                    ; Mera delay?
                BNE  Loop           ; Hoppa om JA

SLUT          STAB  Utport          ; Slut och SLÄCK
                NOP                    ; - eller
                JMP  Start
    
```