

Tentamen i Digital o Datorteknik för E, GU, IT, Z. 2006-04-18

Kortform av lösningar till tentan. För full poäng krävs fullständiga lösningar enligt typtentan

1a) $R=X-Y$ utförs som $R=X+Y_{1k}+1$; $Y_{1komp} = 00110$.

	011111
X	01101
+Y _{1komp}	+ 00110
=R	= 10100

1b) $N=1$; $Z=0$; $V=1$; $C_5=0 \Rightarrow C=1$

1c) $X=13$ $Y=25$; $R=20$ (Kontroll: $13-25 \neq 20$); verkar rimligt ty $C=1$
 C anger att resultatet är fel vid tal utan tecken

1d) $X=13$; $Y=-7$; $R=-12$ (Kontroll: $13-(-7) \neq -12$); verkar rimligt ty $V=1$
 V anger fel vid tal med tecken.

1e) Jämn paritet. Kodordet CB_{16} innehåller ett udda antal ettor vilket innebär fel. Kodorden motsvarar texten "Kert". Svenskt namn kan vara "Kurt". ASCII för "e" = 1100101. ASCII för "u" = 1110101. Troligen är b_5 i kodordet ändrat.

Upp 2

2a) Enligt tabellen är $g(xyz) \neq f(xyz)$

xyz	$y \oplus z$	xz	f	$(x'+z')$	$(y'+z)$	g
000	0	0	0	1	1	1
001	1	0	1	1	1	1
010	1	0	1	1	0	0
011	0	0	0	1	1	1
100	0	0	0	1	1	1
101	1	1	1	0	1	0
110	1	0	1	1	0	0
111	0	1	1	0	1	0

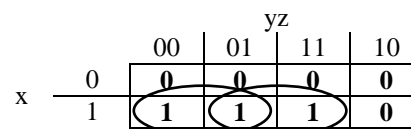
2b) Se blåa boken del 1 exempel 5.13

2c)

xyz	F
000	0
001	0
010	0
011	0
100	1
101	1
110	0
111	1

Disjunktiv minimal form:
 $f = (xy') + (xz)$

Konjunktiv minimal form:
 $f = (x)(y'+z)$



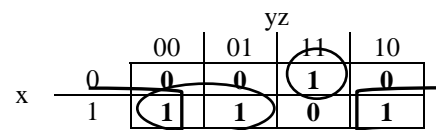
2d) Rita nätet

Upp 3 a)

xyz	F
000	0
001	0
010	0
011	1
100	1
101	1
110	1
111	0

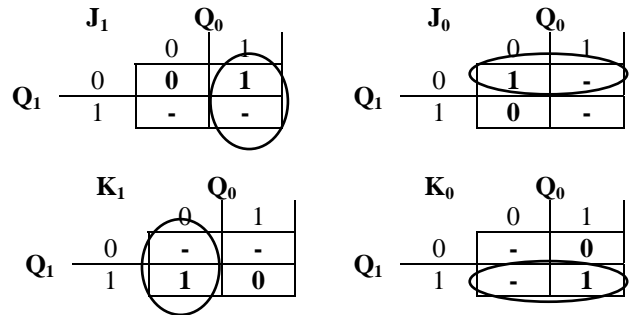
Minimerat blir $f = (xy') + (xz') + (x'yz)$

Rita nätet med NAND/NAND-logik



Upg 3b)

Detta Tillst	Nästa tillst		
$q_1 q_0$	$q_1^+ q_0^+$	$J_1 K_1$	$J_0 K_0$
00	01	0 -	1 -
01	11	1 -	- 0
10	00	- 1	0 -
11	10	- 0	- 1



Rita figur med följande insignaler till vipporna

$J_1 = q_0$	$J_0 = q_1'$
$K_1 = q_0'$	$K_0 = q_1$

Upg 4

4a)

State nr	RTN-beskrivning	Styrsignaler (=1)
0	PC → MA, PC+1 → PC, S-1 → S	OE _{PC} , LD _{MA} , IncPC, DecS
1	M → T	MR, LD _T
2	S → MA	OE _S , LD _{MA}
3	PC → M, T → R	OE _{PC} , MW, f ₁ , LD _R
4	R → PC, NF	OE _R , LD _{PC} , NF

4b)

- 0) Förbered för läsning av adressoperand i minnet, Öka PC med ett, Minska stackpekaren
- 1) Läs adressoperanden från minnet till register T
- 2) Förbered för att spara PC
- 3) Spara PC på stacken, Flytta adressoperanden till R
- 4) Och vidare till PC, Ny Fetch

Instruktionen är JSR \$Adr

4c)

State nr	RTN-beskrivning	Styrsignaler (=1)
0	PC → MA, PC+1 → PC	OE _{PC} , LD _{MA} , IncPC
1	M → MA	MR, LD _{MA}
2	M → A, NF	MR, LD _A , NF

Upg 5

- 5a) PC är 8 bitar $2^8=256$ adresser. Databussen är 8 bitar bred; $256*8=2048$. Ant bitar totalt= 2048.
- 5b) Alla siffror = Hexsiffror! Först minskas X, och vi får 3 varv $17+16+15=42$ som skrivs till adr F0
- 5c) ADCA och DEC påverkar båda C-flaggan vilket verkar konstigt. Å andra sidan kan det verka fel att utnyttja ADCA i snurran.

Upp 6a

```

IRQINIT  psha
         pshx
         movw    0,CLOCK      nollställ kolckan
         movb    0,CLOCK+2
         ldaa   #100         Avbrottsräknare
         staa   TEMP
         staa   IRQRES      nollställ avbrottsvippan
         ldx    #IRQ         avbrottsvektor
         stx    $fff2       (alt 3ff2)
         cli
         pulx
         pula
         rts

TEMP     rmb      1          Avbrottsräknare (100 IRQ = 1s)

IRQ      sta      IRQRES     nollställ avbrottsvippan
         dec      TMP        100 avbrott?
         bne      IExit      nej
         ldaa   #100         Avbrottsräknare
         staa   TEMP

* Öka sekunder
         ldaa   CLOCK+2
         adda   #1
         daa
         staa   CLOCK+2
         cmpa  #60           Hel minut?
         bne      IExit      nej

* Öka minuter
         clr     CLOCK+2     Nolla sekunder
         ldaa   CLOCK+1
         adda   #1
         daa
         staa   CLOCK+1
         cmpa  #60           Hel timme?
         bne      IExit      nej

* Öka timmar
         clr     CLOCK+1     Nolla minuter
         ldaa   CLOCK
         adda   #1
         daa
         staa   CLOCK
         cmpa  #24           24 timmar?
         bne      IExit      nej
         clr     CLOCK

IExit    rti              (Plus programhuvud och flödesplan)

```

Upp 6b

```

Start    LDX    #SegCode    Pekare till tabell
         LDAB   Inport      Läs inporten
         CMPB  #10         Giltigt värde
         BLO   OK          ..hoppa om JA
         LDAA  #Error      Skriv Error
         STAA  Utport
         BRA   End

OK       LDAA  B,X         Översätt indata till Segmentkod
         STAA  Utport      .. och skriv ut

End      BRA   Start

```

Upg 7 – för D-linjen

- 7. a) Synkront sekvensnät
- b) Tillstånds- och utsignalstabellen

Ur kopplingen kan vi teckna de Booleska uttrycken för q_1^+ , q_0^+ och u :

$$q_1^+ = x_2'x_1x_0'q_0$$

$$q_0^+ = x_2'x_1x_0$$

$$u = (x_2x_1'x_0'q_1)'$$

Tillstånd	Insign	Nästa tillst	Utsign	
q_1q_0	$x_2x_1x_0$	$q_1^+q_0^+$	u	
τ_0	0 0	0 1 0	0 0	1
		0 1 1	0 1	1
		1 0 0	0 0	1
		övrr	0 0	1
τ_1	0 1	0 1 0	1 0	1
		0 1 1	0 1	1
		1 0 0	0 0	1
		övrr	0 0	1
τ_2	1 0	0 1 0	0 0	1
		0 1 1	0 1	1
		1 0 0	0 0	0
		övrr	0 0	1
τ_3	1 1	0 1 0	1 0	1
		0 1 1	0 1	1
		1 0 0	0 0	0
		övrr	0 0	1

c) Funktionsbeskrivning i ASM-plan

