

Tentamen i
Digitala system - EITA15 15hp
varav denna tentamen 4,5hp

Institutionen för elektro- och informationsteknik
Campus Helsingborg, LTH

2018-01-09 8.00 - 13.00 (förlängd 14.00)

Uppgifterna i tentamen ger totalt 60 poäng. Uppgifterna är inte ordnade på något speciellt sätt. Läs därför igenom alla uppgifter innan du börjar lösa dem. Några uppgifter är uppdelade i deluppgifter. Av totalt 60 möjliga poäng fordras minst 30 för godkänt.

Betygsgränser:

- 30p - 39p ger betyg 3
- 40p - 49p ger betyg 4
- 50p - 60p ger betyg 5

Inga hjälpmedel är tillåtna

Observera!

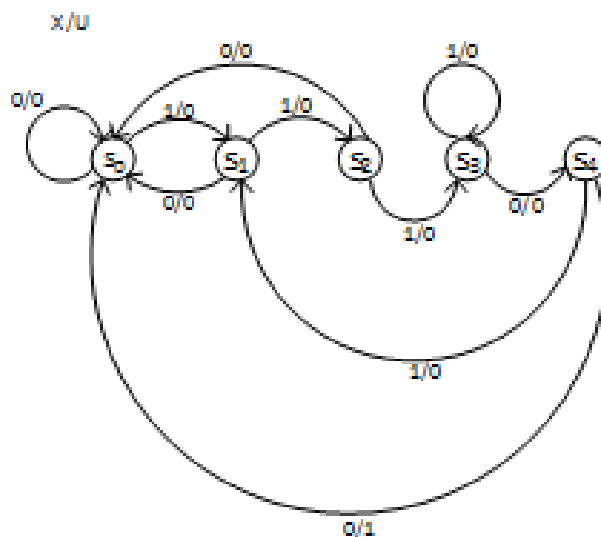
- För att rättning av lösning skall komma i fråga fordras att den är läslig samt klart och tydligt uppställd.
- Glöm inte att skriva personlig identifierare på varje blad.
- Alla lösa blad ska vara samlade i omslaget.
- Lösningarna ska vara numrerade och ordnade i nummerföljd.
- Påbörja ny uppgift på nytt papper.

Lycka till!

1. Denna uppgift består av fem påstående. Besvara varje påstående om det är sant eller falskt. Varje rätt påstående ger +2 poäng medan fel svar ger -1 poäng. Maximala antalet poäng på denna uppgift är 10 och minsta poäng 0. (inga minuspoäng totalt!)

- (a) falsk
- (b) sant
- (c) falskt
- (d) falskt
- (e) sant

2. (a) kodning $s_0=000$, $s_1=001$, $s_2=011$, $s_3=010$, $s_4=110$



(b) SP-form

q2 q1 q0	x=0	x=1
0 0 0	000/0	001/0
0 0 1	000/0	011/0
0 1 1	000/0	010/0
0 1 0	110/0	010/0
1 1 0	000/1	001/0

		q_0x			
		00	01	11	10
q_2q_1	00	0	0	0	0
	01	1	0	0	0
	11	0	0	-	-
	10	-	-	-	-

		q_0x			
		00	01	11	10
q_2q_1	00	0	0	1	0
	01	1	1	1	0
	11	0	0	-	-
	10	-	-	-	-

		q_0x			
		00	01	11	10
q_2q_1	00	0	1	1	0
	01	0	0	0	0
	11	0	1	-	-
	10	-	-	-	-

q_2^+

q_1^+

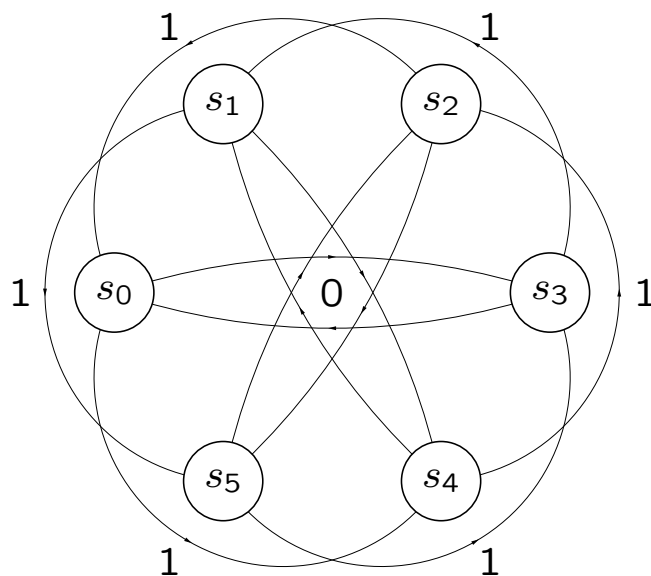
		q_0x			
		00	01	11	10
q_2q_1	00	0	0	0	0
	01	0	0	0	0
	11	1	0	-	-
	10	-	-	-	-

q_0^+

u

$$\begin{aligned}
 q_2^+ &= x'q_0'q_1q_2' \\
 q_1^+ &= xq_0 + q_0'q_1q_2' \\
 q_0^+ &= xq_1' + xq_2 \\
 u &= x'q_2
 \end{aligned}$$

3. Modulo 6



Tabell:

s	0	1
s_0	s_3	s_4
s_1	s_4	s_5
s_2	s_5	s_0
s_3	s_0	s_1
s_4	s_1	s_2
s_5	s_2	s_3

Koda i NBCD

s	Tillstånd $q_1q_2q_3$	x	
		0	1
s_0	000	011	100
s_1	001	100	101
s_2	010	101	000
s_3	011	000	001
s_4	100	001	010
s_5	101	010	011

		q_0x			
		00	01	11	10
q_2q_1	00	0	1	1	1
	01	1	0	0	0
	11	-	-	-	-
	10	0	0	0	0

		q_0x			
		00	01	11	10
q_2q_1	00	1	0	0	0
	01	0	0	0	0
	11	-	-	-	-
	10	0	1	1	1

q_2^+

q_1^+

		q_0x			
		00	01	11	10
q_2q_1	00	1	0	1	0
	01	1	0	1	0
	11	-	-	-	-
	10	1	0	1	0

q_0^+

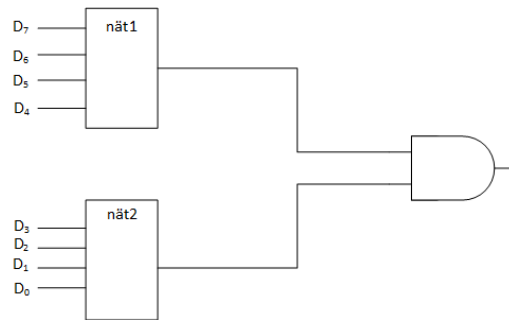
minimal SP-form:

$$q_2^+ = q_1q_0'x' + q_2q_1'x + q_2q_1'q_0$$

$$q_1^+ = q_2x + q_2q_0 + q_2'q_1'q_0'x'$$

$$q_0^+ = q_0'x' + q_0x$$

4. Om vi delar upp uppgiften i två delar, nät 1 och nät2. Nät1 hanterar de 4 mest signifikanta bitarna och nät2 de fyra minst signifikanta. Nät1 ska ge etta ut då insignalen är lika med 3 och nät2 ska ge etta ut då insignalen är mellan 0 och 9.



		D_1D_0			
		00	01	11	10
D_3D_2	00	1	1	1	1
	01	1	1	1	1
	11	0	0	0	0
	10	1	1	0	0

		D_5D_4			
		00	01	11	10
D_7D_6	00	0	0	1	0
	01	0	0	0	0
	11	0	0	0	0
	10	0	0	0	0

nät2

nät1

$$\text{nät1} = D_5D_4D'_6D'_7$$

$$\text{nät2} = D'_3 + D'_1D'_2$$

5. (a) För varje instruktion 1, 2, 3, och 4 är tiderna:
 Instruktion 1 (LOAD R1, B) tar: $100+1=101\text{ns}$
 Instruktion 2 (LOAD R2, C) tar: $100+1=101\text{ns}$
 Instruktion 3 (ADD R1, R2) tar: $1+1+1+1=4\text{ns}$
 Instruktion 4 (STORE A, R1) tar: $100+1=101\text{ ns}$
 Den totala tiden är summan, dvs: $101+101+4+101=307\text{ns}$
- (b) Om kompilatorn använt direkt adressering skulle adressen till operanden ligga i instruktionen (Se föreläsning 2). Om processorn tillåter flera operander att vara direkt adresserade skulle följande instruktion räcka:

ADD A, B //A \leftarrow A + B

Instruktionen skulle ta följande tid: 100 ns för att läsa in A, 100 ns för att läsa B, 1 ns för att beräkna additionen, och 100 ns för att skriva resultatet till minnet. Total tid: $100+100+1+100=301\text{ ns}$.

- (c) Programmet skulle kunna se ut så här:

LOAD	R1, B	//R1 \leftarrow B
LOAD	R2, C	//R2 \leftarrow C
LOAD	R3, D	//R1 \leftarrow B
LOAD	R4, E	//R2 \leftarrow C
ADD	R1, R2	//R1 \leftarrow R1 + R2
ADD	R3, R4	//R1 \leftarrow R1 + R2
SUB	R1, R3	//R1 \leftarrow R1 - R3
STORE	A, R1	// A \leftarrow R1

6. (a) Eftersom $2^6 = 64$ så behövs 6 bitar för att specificera adressrymden.
- (b) Lokalitet bestäms av det program som exekveras på processorn. Om det inte finns några loopar eller några hopp, d v s alla instruktioner exekveras en gång så finns det ingen lokalitet.
- (c) Det finns två typer av lokalitet och det är temporal lokalitet och rumslokalitet
- (d) För temporal lokalitet så är det så att om en instruktion/data blivit refererat nu, så är sannolikheten stor att samma referens görs inom kort. Och för rumslokalitet är det så att om instruktion/data blivit refererat nu, så är sannolikheten stor att instruktioner/data vid adresser i närheten kommer användas inom kort
- (e) Om Write-through används så görs skrivningar i cache görs skrivningar direkt i primärminnet. Om Write (Copy)-back används så uppdateras primärminnet först när en cacherad byts ut.