

Grundläggande Datorteknik

Digital- och datorteknik

Välkommen!

Grundläggande Datorteknik fo16

1

Kursens mål:

- Fatta hur en dator är uppbyggd (HDW)
- Fatta hur du programmerar den (SW)
- Fatta hur HDW o SW samverkar

Digital teknik

Dator teknik

Grundläggande Datorteknik fo16

2

Lärandemål:

1) Talsystem, binära koder och datoraritmetik

1. Konvertering mellan olika talsystem
2. Utifrån given problemställning applicera binära koder så som NBC, NBCD, ASCII, Gray, Excess, tecken/belopp och komplementkoder.
3. Redogöra för och tillämpa binär aritmetik (addition och subtraktion).

Grundläggande Datorteknik fo16

3

Talsystem
Basen 2, 8, 10, 16

Positionssystem
Ex: 214
(421)

$\beta = 2$	$\beta = 8$	$\beta = 10$	$\beta = 16$
binärt	oktalt	decimalt	hexa-decimalt
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11

S2.4

Grundläggande Datorteknik fo16

4

GRAY-KOD

Tabell 2.2. Graykoder.

S2.16

Decimal ordning	Kodord i trebitars Graykod	Kodord i fyrbitars Graykod
0	000	0000
1	001	0001
2	011	0011
3	010	0010
4	110	0110
5	111	0111
6	101	0101
7	100	0100
8		1100
9		1101
10		1111
11		1110
12		1010
13		1011
14		1001
15		1000

Grundläggande Datorteknik fo16

NBCD-kod

Skriv (563,782) på NBCD-kod.

Decimal siffra	NBCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Grundläggande Datorteknik fo16

S2.19

Alfanumeriska koder

ASCII-koden	0	1	2	3	4	5	6	7	$b_3b_2b_1b_0$
NUL	DLE	SP	@	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	{	1 0 1 1
FF	FS	,	<	L	\	ö	l	}	1 1 0 0
CR	GS	-	=	M]	å	m	~	1 1 0 1
S0	RS	.	>	N	^	n	-		1 1 1 0
S1	US	/	?	O	_	o	RUBOUT (DEL)		1 1 1 1

S2.29

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

Ex "A" = 41₁₆

Tabell 2.1. Excess-2ⁿ⁻¹ kodning vid n = 4.

Excess-kod

Ex avkoda:
1 1 0 0₂ Excess-2ⁿ⁻¹

n=4

$$2^3 + 2^2 + 0 + 0 - 2^3 = 4_{10}$$

Nivå k ⁹	Kodord i excess-2 ⁿ⁻¹ -kod (n=4)
-8 ⁹	0000
-7 ⁹	0001
-6 ⁹	0010
-5 ⁹	0011
-4 ⁹	0100
-3 ⁹	0101
-2 ⁹	0110
-1 ⁹	0111
0	1000
1 ⁹	1001
2 ⁹	1010
3 ⁹	1011
4 ⁹	1100
5 ⁹	1101
6 ⁹	1110
7 ⁹	1111

Grundläggande Datorteknik fo16

S2.15

8

Def 2-Komplement:

Arb s 30

Pos: $Y = Y$

Neg: $(-Y) = 2^n - |Y| = Y_{2K}$

Att tvåkomplementera:

Ex 4 bit: $2^n = 2^4 = 16$

$$2^n - Y = 2^n - 1 - Y + 1 = 16 - 1 - Y + 1 = 15 - Y + 1 (= Y_{1K} + 1)$$

Ex $Y=6$: 0110 Hitta $(-Y)$

$$\begin{array}{r} 15_{10} \quad 1111 \\ -Y \quad -0110 \\ \hline \end{array}$$

$$= Y_{1K} = 1001$$

INVERSEN! Def 1komp! Y_{1K}

$$\text{addera } 1 \quad +0001$$

$$= Y_{2K} = 1010$$

$$Y_{1K} + 1 = Y_{2K} \quad (-Y = 1010) \\ (-6 = 1010)$$

Att subtrahera:

$$X - Y = X + Y_{2K} = X + Y_{1K} + 1$$

Grundläggande Datorteknik fo16

9

Lärandemål:

2) Digitalteknik

1. Definiera grundläggande logiska operationer och dess motsvarande logiska grindar.
2. Tillämpa den booleska algebras räknelagar.
3. Utföra algebraisk förenkling av booleska uttryck.
4. Visa likhet/olikhet mellan booleska uttryck.

Grundläggande Datorteknik fo16

11

Def Flaggor

Arb s 32

Statusflaggor ut från ALU:n som indikerar om resultatet blev rätt eller fel

C Carry Tal utan tecken [0,15] (ADD : minnessifra; SUB: lånesifra)

V Overflow Tal med tecken [-8,7]

N Negative Tal med tecken [-8,7]

Z Zero Tal med och utan tecken

C=1: Resultatet av operationen blev *fel* för en operation *utan tecken*

V=1: Resultatet av operationen blev *fel* för en operation *med tecken*

N=1: Resultatet av operationen blev *negativt* för en operation *med tecken*

Z=1: Resultatet av operationen blev *noll*

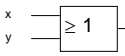
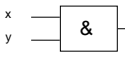
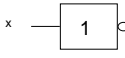
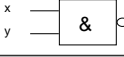
Grundläggande Datorteknik fo16

10

Logikkretssymboler för grundläggande logikoperationer.

S3.18

S1.14

Funktion	Grind	Grafisk symbol
$z = x + y$	ELLER (OR)	
$z = x \cdot y$	OCH (AND)	
$z = x'$	INVERTERARE (ICKE, NOT)	
$z = (x + y)'$	NOR	
$z = (x \cdot y)'$	NAND	

Grundläggande Datorteknik fo16

12

Sammanfattning

•Boolesk algebra

$1+0=1; 1\cdot 0=0; 1+1=1; 1\cdot 1=1; 1'=0; 0'=1$
de Morgans lagar

•Funktionstabell

Sätt upp tabell med alla kombinationer av invariabler

•Binär evaluering

Gör kolumner för deluttrycken

Rep: Satser inom Boolesk algebra.

S3.8

1. Kommutativa lagarna

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

2. Distributiva lagarna

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

7. Associativa lagarna

$$x + (y + z) = (x + y) + z$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

8. De Morgans lagar

$$(x + y)' = x' \cdot y'$$

$$(x \cdot y)' = x' + y'$$

3.

$$x + 0 = x$$

$$x \cdot 1 = x$$

4.

$$x + x' = 1$$

$$x \cdot x' = 0$$

5.

$$x + 1 = 1$$

$$x \cdot 0 = 0$$

6.

$$x + x = x$$

$$x \cdot x = x$$

9.

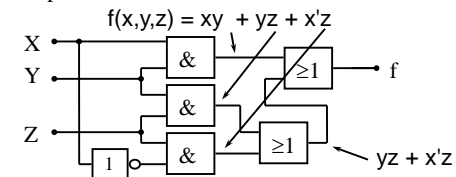
$$(x')' = x$$

Lärandemål:

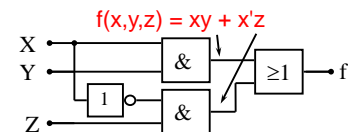
2) Digitalteknik - Kombinatoriska nät

1. Realisera logiska uttryck med grindnät.
2. Beskriva, analysera och konstruera kombinatoriska nät med hjälp av funktionstabeller och boolesk algebra.
3. Kunna minimera logiska uttryck för realisering i kombinatoriska nät.

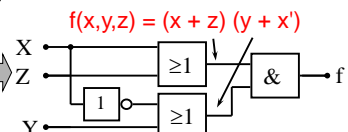
Grindnät för exempel Kalle



Disjunktiv (minimal) form
(Summa av Produkter)



Konjunktiv (minimal) form
(Produkt av summor)



Sammanfattning

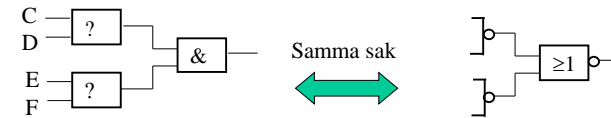
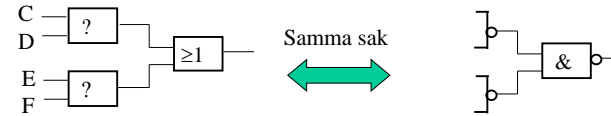
- NORMAL form \Rightarrow Funktionstabell
- MINIMAL form \Rightarrow Karnaughdiagram

- DISJUNKTIV (*normal / minimal*) form
 - Σ av Prod Ex: $(x'y)+(xw)+(xyw)$
 - Ettor
 - Mintermer: $(1 \cdot 1 \cdot 1) = 1$
 - NAND / NAND - logik

DSP eller LEEDS P

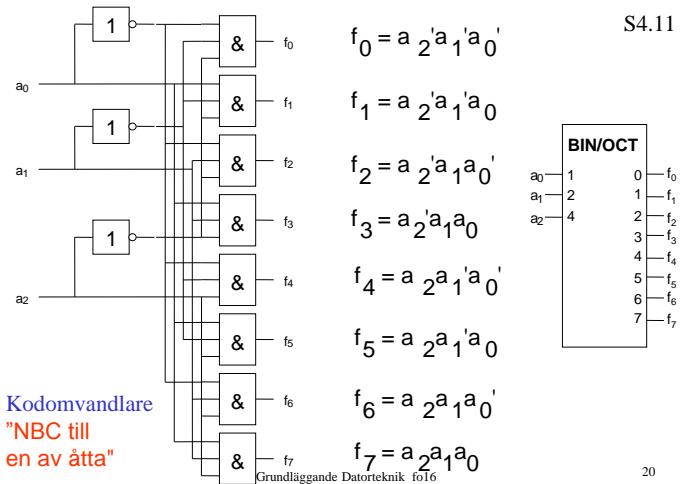
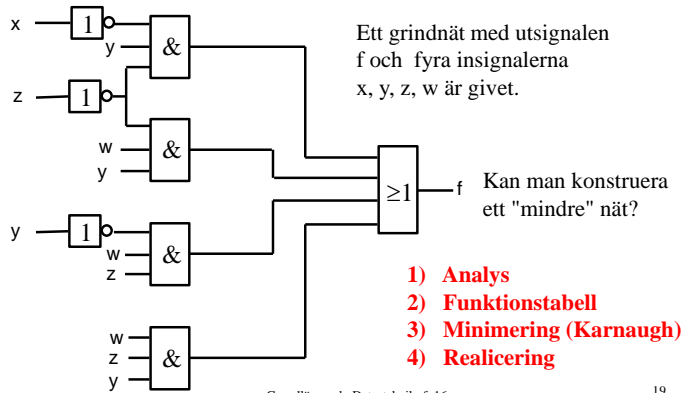
- KONJUNKTIV (*normal / minimal*) form
 - Prod av Σ :or Ex: $(x+z)(x'+z+w)(z'+w')$
 - Nollor
 - Maxtermer: $(0+0+0) = 0$
 - NOR / NOR - logik

Sammanfattning NAND- och NOR-logik



Praktikfall, minimering av grindnät

Ext4



En delmängd av veckans mål:

Fo4

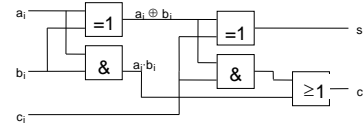
- ▶ Konstruera de olika kombinatoriska nät som ingår i en dator. Exempel på sådana nät är väljare, kodomvandlare och ALU (beräkningseenheten i processorn).
- ▶ Studera hur addition/subtraktion utförs

Dagens mål:

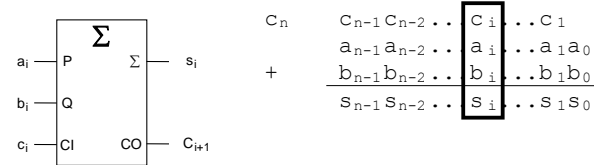
- ▶ Kodomvandlare (en kod IN → annan kod UT)
- ▶ Don't care – termer (ger färre grindar)
- ▶ Väljare (många signaler IN + styrsig → en signal UT)
- ▶ Fördelare (en signal IN+styrsig → många signaler UT)
- ▶ Heladderare (adderar $x+y+c_{in}=s_{ut}$ och c_{ut})
- ▶ Koda tal (2-komplementsrepresentationen)

**Läs mindre!
Lär dig mer!**

S 4.5



Prosamsymbolen för en heladderare.



Def Flaggor

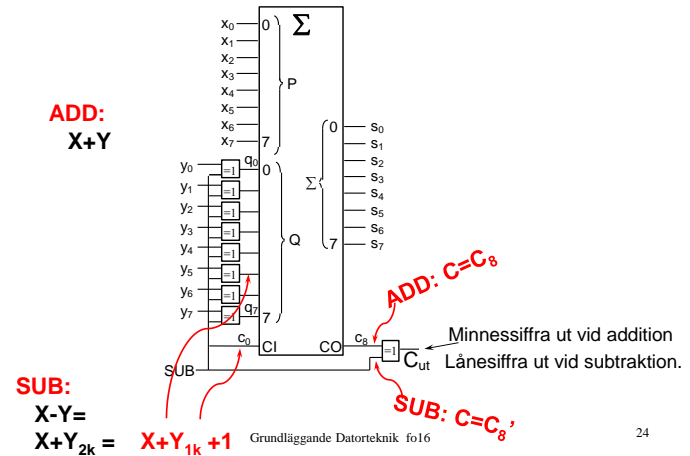
Arb s 32

Statusflaggor ut från ALU:n som indikerar om resultatet blev rätt eller fel

- C** Carry Tal utan tecken [0,15] (ADD : minnessiffra; SUB: lånesiffra)
- V** Overflow Tal med tecken [-8,7]
- N** Negative Tal med tecken [-8,7]
- Z** Zero Tal med och utan tecken

- C=1:** Resultatet av operationen blev *fel* för en operation *utan tecken*
- V=1:** Resultatet av operationen blev *fel* för en operation *med tecken*
- N=1:** Resultatet av operationen blev *negativt* för en operation *med tecken*
- Z=1:** Resultatet av operationen blev *noll*

Ext6

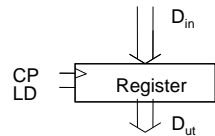
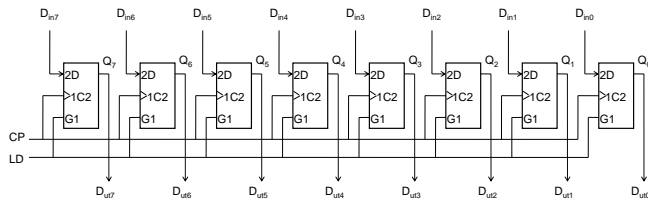


Lärandemål:

2) Digitalteknik - Sekvensnät

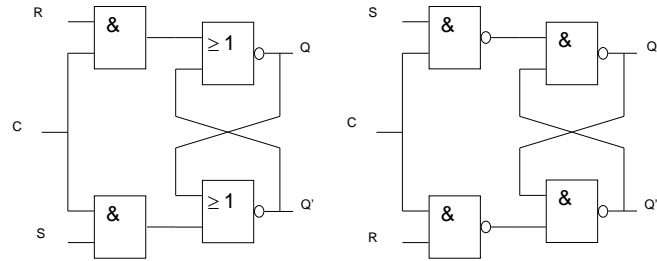
1. Analysera och konstruera synkrona tillståndsmaskiner med hjälp av tillståndstabeller och tillståndsgrafer.
2. Använda D-, T- och JK- vippor för konstruktion av minneselement och räknare.

Register och bussar

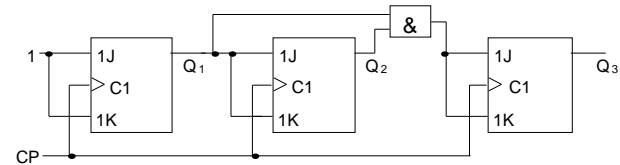


Grundad SR-latch

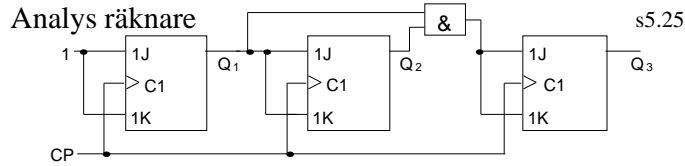
Ofta förses SR-latchar med en tredje ingång, till vilken en styrpuls C ansluts. Härvid erhålls en så kallad grundad SR-latch.



Arbetsgång - analys räknare s5.25 ->



- 1 Studera kopplingen och **bestäm vippornas insignaler** ($T_1=$, $T_2=$, $T_3=$)
- 2 Sätt upp en tabell med
 - "**Detta tillstånd**" (Alla kombinationer av Q_1 , Q_2 , Q_3)
 - **Insignaler** (T_1 , T_2 , T_3)
 - "**Nästa tillstånd**" (Q_1^* , Q_2^* , Q_3^*)
- 3 Ange insignalernas värden i tabellen och----
- 4 ange vad "nästa tillstånd" blir
- 5 Rita slutligen en **tillståndsgraf**



1)

$$T_1=1$$

$$T_2=Q_1$$

$$T_3=Q_1 Q_2$$

Funktionstabell

T	Q'
0	Q
1	Q'

2)

Detta Tillstånd			Insignaler			Nästa Tillstånd		
Q ₃	Q ₂	Q ₁	T ₃	T ₂	T ₁	Q ₃ ⁺	Q ₂ ⁺	Q ₁ ⁺
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	1	1	1	1	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	1	1	1	0	0	0

Grundläggande Datorteknik fo16

3)

4)

Utsignaler								Insignaler							
Detta tillstånd Q				Nästa tillstånd Q'				J ₃	K ₃	J ₂	K ₂	J ₁	K ₁	J ₀	K ₀
q ₃	q ₂	q ₁	q ₀	q ₃ ⁺	q ₂ ⁺	q ₁ ⁺	q ₀ ⁺								
0	0	0	0												
0	0	0	1												
0	0	1	0												
0	0	1	1												
0	1	0													
0	1	1													
0	1														
0		1													
0			1												
1	0	1	0												
1	0	1	1												
1	1	0	0												
1	1	0	1												
1	1	1	0												
1	1	1	1												

Grundläggande Datorteknik fo16

Arbetsgång - syntes räknare

Konstruera en räknare som räknar sekvensen ????

- 1 Rita en **tillståndsgraf**
- 2 Sätt upp en **tabell** med:
 - "Detta tillstånd" (Alla kombinationer av Q₁, Q₂, Q₃)
 - "Nästa tillstånd" (Q₁⁺, Q₂⁺, Q₃⁺)
 - Vippornas **Insignaler**
- 3 Ange "**Nästa tillstånd**" i tabellen
- 4 Använd vippornas excitationstabell och ange **vippornas insignaler**
- 5 **Minimera** uttrycken för insignalerna
- 6 **Realisera** räknaren