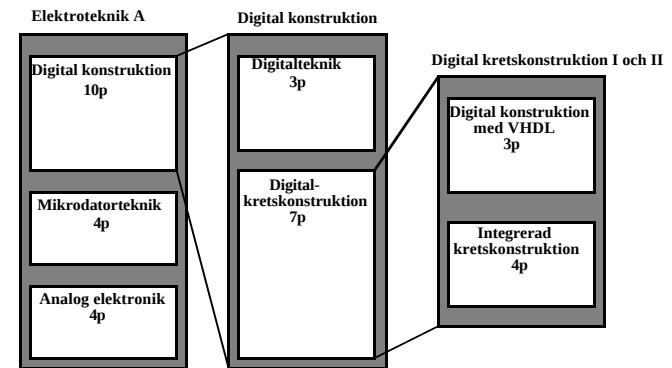


Digitalteknik 3p, vt 01

- **Kurslitteratur:** "A First Course in Digital Systems Design - An Integrated Approach"
- **Antal föreläsningar:** 11 (2h)
- **Antal laborationer:** 4 (4h)
- **Examinationsform:** Tentamen + laborationer
- **Kursansvarig:** Bengt Oelmann
- **Övningsassistent:** Bengt Oelmann
- **Laborationshandledare:** Bengt Oelmann

1 (20)

Struktur: Elektroteknik A



2 (20)

Motivation och målsättning för kurserna i digital elektronik

- **Digitalteknik, 3p**
 - Ge grundläggande kunskaper som bas för kommande konstruktionskurser
 - Switching algebra, formell analysteknik för digitala kretsar
 - Procedurer för att ta fram och manipulera kombinatoriska nät
 - Procedurer för att ta fram sekvensnät
- **Digitalteknik I, 3p**
 - Konstruktionskurs med moderna konstruktionsmetoder
 - Grundläggande byggblock i digitala system
 - Hårdvarubeskrivande språk (VHDL) för simulering och syntes
- **Digitalteknik II, 4p**
 - Konstruktionskurs med inriktning på integrerad krets konstruktion
 - Implementationstekniker (full-custom, standard cell, programmerbara kretsar)
 - analys och optimering av prestanda av IC
 - Konstruktionsmetoder och verktyg

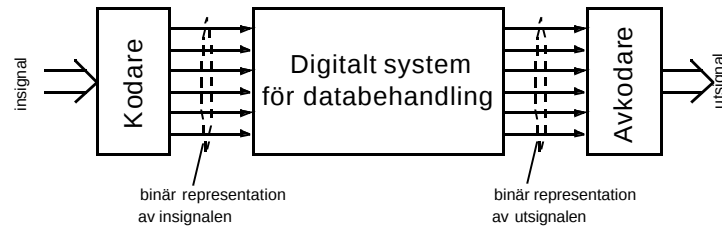
3 (20)

F1: Introduktion

- **Innehåll:**
 - Representation av data i digitala system
 - Grundläggande funktioner i digitala system

4 (20)

Vad är ett digitalt system ?



- Ett digitalt system är ett elektroniskt nät som behandlar information genom att göra beräkningar och operationer på enbart tal
 - Ett digitalt system använder sig av det binära talsystemet
 - Ett digitalt system måste hantera följande:
 - Översätta information från den "riktiga världen" till ett binärt format (kodning)
 - Utföra beräkningar och operationer på binära tal
 - Returnera ett svar tillbaks till den "riktiga världen" (avkodning)

5 (20)

Binära tal

- En binär siffra kan anta värdena 0 och 1 (en bit)
- En binär variabel kan anta värdena 0 och 1

Exempel: definiera den binära variabeln A av en bit
Följande tilldelningar är tillåtna: A = 0 och A = 1

- Ett binärt tal kan vara sammansatt av en grupp av bitar

Exempel: definiera den binära variabeln B av 3 bitar
 $B = b_2 b_1 b_0$
B kan anta ett av 8 olika värden. B = 000, 001, 010, 011, 100, 101, 110, 111

- Generellt
en variabel av n-bitar (där n är ett heltal >0) kan anta 2^n olika värden

6 (20)

Talsystem och digitala koder

- Digitala system är byggda av kretsar som endast hanterar binära siffror, 0 och 1.
- Verkliga problem som digital system ska hantera är sällan baserade på binära siffror.
- Digitalkonstruktören måste finna samband mellan binära siffror i kretsen och värden som finns i verkligheten
 - numeriska data (mätdata som t.ex. temperatur)
 - händelser (indikera att en tangent är nedtryckt)
 - tillstånd (telefonlinjen är upptagen)
 - villkor (talet a är större än talet b)

7 (20)

Positionsbaserade talsystem

- Exempel 1: decimalt heltal
- $D = 4374 = 4 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$
($d_3 d_2 d_1 d_0$) = 4374; $r = 10$; $p = 4$; $n = 0$
- Exempel 2: decimalt tal med decimalpunkt
- $D = 0.454 = 0 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} + 4 \times 10^{-3}$
($d_3 d_2 d_1 d_0$) = 0454; $r = 10$; $p = 1$; $n = 3$

- Generellt

d_{p-1}	d_{p-2}	d_1	d_0	.	d_{-1}	d_{-2}	$d_{-(n-1)}$	d_{-n}
$\times r^{p-1}$	$\times r^{p-2}$	$\times r^1$	$\times r^0$		$\times r^{-1}$	$\times r^{-2}$	$\times r^{-(n-1)}$	$\times r^{-n}$

$$D = \sum_{i=-n}^{p-1} d_i r^i$$

8 (20)

Representation av tal i digitala system

• Exempel

$$10101_2 = 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 21_{10}$$

$$00010_2 = 0 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 2_{10}$$

$$01.01_2 = 0 \times 2 + 1 \times 1 + 0 \times 0.5 + 1 \times 0.25 = 2.75_{10}$$

• Generellt

Ett binärt tal skrivs som:

b_{p-1}	b_{p-2}	b_1	b_0	.	b_{p-1}	b_2	$b_{(n-1)}$	b_n
$\times 2^{p-1}$	$\times 2^{p-2}$	$\times 2^1$	$\times 2^0$		$\times 2^{-1}$	$\times 2^{-2}$	$\times 2^{-(n-1)}$	$\times 2^{-n}$

$$B = \sum_{i=-n}^{p-1} b_i 2^i$$

- siffran längst till vänster kallas för *den mest signifikanta biten* MSB (Most Significant bit)
- siffran längst till höger kallas för *den minst signifikanta biten* LSB (Least Significant bit)
- den binära motsvarigheten till decimalpunkt kallas för *binary radix point*

Oktala och hexadecimala tal

• Motivation för att använda olika baser:

- Basen 10: används till vardags
- Basen 2: digitala system består av kretsar som bara kan hantera två värden
- Basen 8,16: Ger ett kompakt skrivsätt då man arbetar med digitala system

• Oktala tal (basen 8)

- varje position representerar ett tal mellan 0 och 7
- en sträng med tre bitar representeras med en oktal siffra (0-7)
- Exempel på konvertering från binärt till oktal:

$$10\ 111\ 101\ 010_2 = 2752_8$$

• Hexadecimala tal (basen 16)

- varje position representerar ett tal mellan 0 och 15
- en sträng med fyra bitar representeras med en hexadecimal siffra (0-9,A-F)
- Exempel på konvertering från binärt till hexadecimalt:

$$0010\ 0011\ 1011\ 1111 = 23BF_{16}$$

Konvertering av tal mellan olika baser

• Vid kovertering från en bas till en annan,

- Låt värdet av talet (talet som vi vill omvandla från) vara D

Värdet för ett tal D med basen r kan skrivas som:

$$D = \sum_{i=-n}^{p-1} d_i r^i$$

en omskrivning av summan ger, då $n=0$:

$$D = d_0 + d_1 \cdot r + \dots + d_{p-2} \cdot r^{p-2} + d_{p-1} \cdot r^{p-1}$$

$$D = (((d_{p-1} \cdot r + d_{p-2}) \cdot r + \dots) \cdot r + d_1) \cdot r + d_0$$

Exempel: konvertering hexadecimal till decimal

$$A3F7_{16} = ((10 \times 16 + 3) \times 16 + 15) \times 16 + 7 = 41975_{10}$$

Konvertering från decimalt till basen r

Återigen skriver vi värdet för ett tal D med basen r som:

$$D = (((d_{p-1} \cdot r + d_{p-2}) \cdot r + \dots) \cdot r + d_1) \cdot r + d_0$$

genom att utföra divisionen $Q=D/r$ får vi:

kvoten:

$$Q = ((d_{p-1} \cdot r + d_{p-2}) \cdot r + \dots) \cdot r + d_1$$

och resten: d_0 som är den minst signifikanta siffran i basen r .

Kvoten Q är på samma form som D . Genom repetition kan alla siffror tas fram.

Exempel:

Konvertera 6_{10} till basen 2 (binärt).

$$6 \div 2 \rightarrow \text{kvoten} = 3 \text{ och resten} = 0 \text{ (LSB)}$$

$$3 \div 2 \rightarrow \text{kvoten} = 1 \text{ och resten} = 1$$

$$1 \div 2 \rightarrow \text{kvoten} = 0 \text{ och resten} = 1 \text{ (MSB)}$$

$$\text{Svar: } 6_{10} = 110_2$$

- För decimaldel får man multiplicera med basen istället.

Omvandling från decimal till basen r

- **Procedur för omvandling av bråkdel**

```
binary dec2bin(float x) {
    int i = -1;
    binary b;
    while(x) {
        x *= 2;
        if(x>1) x -= 1;
        b(i--) = x>1 ? 1 : 0;
    }
    return b;
}
```

- **Procedur för omvandling av heltal**

```
binary dec2bin(int x) {
    int i = 0;
    binary b;
    while(x) {
        b(i++) = x mod 2;
        x = x div 2;
    }
    return b;
}
```

Representation av negativa tal

- **Teckenbit (signed-magnitude)**

sign-bit	b_7	b_6	b_5	b_4	...	b_1	b_0
----------	-------	-------	-------	-------	-----	-------	-------

En extra bit (sign-bit) införs för att representera vilket tecken talet har.

Positivt tal: sign-bit = 0

Negativt tal: sign-bit = 1

n -bit tal kan representera värden mellan $-(2^{n-1}-1)$ och $+(2^{n-1}-1)$ med två sätt att representera 0.

- **Exempel**

$01011101_2 = 93_{10}$
 $00000000_2 = +0_{10}$

$11011101_2 = -93_{10}$
 $10000000_2 = -0_{10}$

Svårt att implementera logiska kretsar som använder teckenbit.

Två-komplement representation (two's-Complement)

- **MSB fungerar som teckenbit - talet är negativt endast om MSB=1**

Till skillnad från ett tal utan tecken har MSB värdet -2^{n-1} (istället för $+2^{n-1}$).

Tal inom området $-(2^n)$ till $+(2^{n-1}-1)$ kan representeras.

Byt tecken på ett tvåkomplement tal:

- Utför bitvis invertering
- Addera 1

Exempel (8 bitars tal):

$27_{10} = 00011011_2$
 invertera $\rightarrow 11100100$
 $+1$
 $-27_{10} = 11100101_2$

$0_{10} = 00000000_2$
 invertera $\rightarrow 11111111$
 $+1$
 $0_{10} = 00000000_2$

$-27_{10} = 11100101_2$
 invertera $\rightarrow 00011010$
 $+1$
 $27_{10} = 00011011_2$

$-128_{10} = 10000000_2$
 invertera $\rightarrow 01111111$
 $+1$
 $-128_{10} = 10000000_2$

BCD-koder

- **Exempel**

Skriv talen 17345₁₀ på binärt format och BCD-format.

$17345_{10} = 100001111000001_2$

I BCD-kod för varje decimal siffra en 4-bitars strängkod.

1	7	3	4	5
0001	0111	0011	0100	0101

Observera att i binär form krävs 15 bitar medans i BCD-kod krävs 20 bitar.

Tecken koder

- För icke-numeriska data som text krävs speciella koder

Den vanligaste teckenkoden för alfanumeriska data är ASCII (American Standard Code for Information Interchange).

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	^	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	EMQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EN	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	=	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

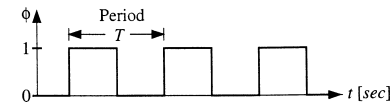
17 (20)

Grundläggande funktioner

- Grundläggande funktioner (eller system primitiver) är sådana som används flera gånger för att skapa ett komplett system.

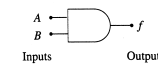
- klocka

är tidsreferensen i digitala system och används för att synkronisera operationerna i systemet



- logiska grindar

en grind utför en logisk funktion ($f = f(A,B)$) där f är grindens utgång och A,B är grindens ingångar



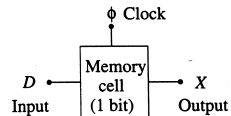
A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

18 (20)

forts. Grundläggande funktioner

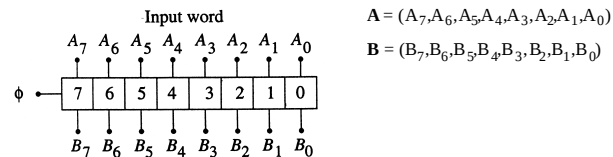
- Minnen

ett minne kan hålla ett binärt värde.



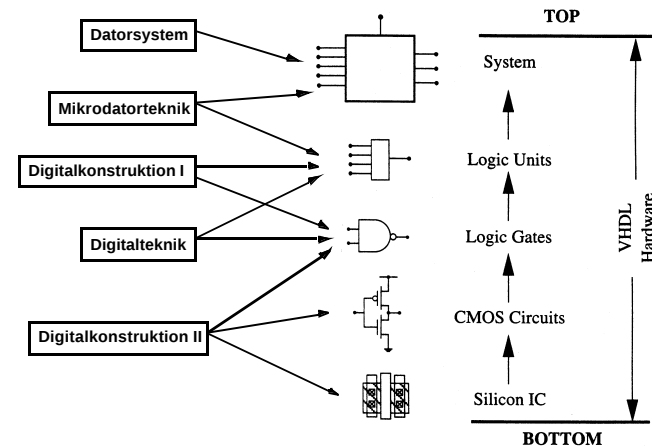
- Register

ett register kan lagra ett binärt tal (8 bitar som exemplet nedan)



19 (20)

Uppbyggnad av kurserna i digital elektronik



20 (20)