Digital och Datorteknik

Välkommen!

Dig o Dat = DoD

LP1 EDA432 (IT1), DIT790 (GU), LEU431 (L)
LP2 EDA215 (Z1), DIT790 (GU), EDA451 (D1)
LP4 EDA311 (E1)

Kursens mål:

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

… och därmed kunna använda en
modern μ-processor / μ-controller

Digital teknik

Dator teknik

Kursens mål:

Dator teknik

Högnivå språk (Java Ada C)
Kompilator (översättare)
Avbrott (realtid)

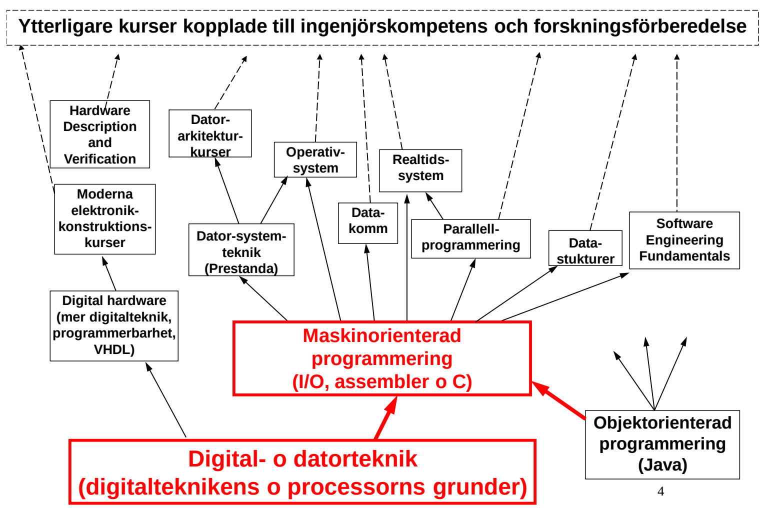
Använda en modern processor
I/O Minne
Maskin kods programmering
Maskin instruktioner

Digital teknik

Kombinatoriska nät
Logik: Register
Vippor
Grindar

Data väg
Binär kodning

Fysik:
Kisel
Transistorer
Elektroniska fenomen



Hur når vi målet? forts

Arb s 7

Beskrivning av funktion

”Automatiskt styrd borrmaskin”
- **Positionera borr**
- **Starta borr**
- **Borra genom arbetsstycke**
- **...**

Assemblerspråk

```
Fortsätt  STAA   BorrStyr
          LDAA   BorrStat
          ANDA   #B1Mask
          CMPA   #BorrNere
          BNE    Fortsätt
          ---
```

Beskrivning av styrsignaler

CP1: OE_{PC}=1, LD_{Adr} =1, Inc_{PC}=1
CP2: MR=1, LD_T=1
CP3: OE_{DR1}=1, LD_R, f_3=1 f_1=1
...

Kursens Hemsida

Sök via studieportalen
EDA432

eller
http://www.cse.chalmers.se/edu/course/EDA432

Länkar till kursdokument
- Senaste nytt
- Kurs-PM
- Veckoplanering
- Laborationsinformation
- Simulatorer

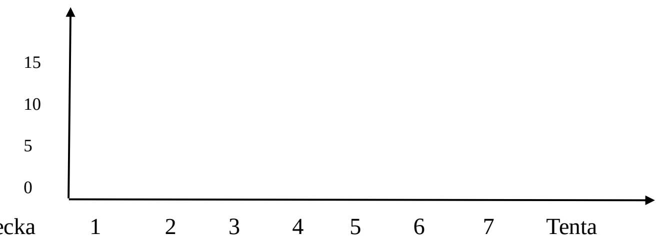
Hur följer du kursen

Läs smart! Lär dig mer!

- Fokusera på:
  - ✓Kursens mål
  - ✓Veckans mål
  - ✓Föreläsningens mål
- Förbered dig innan föreläsningarna:
  - ✓Använd Veckoplanering
  - ✓ Veckans OH-bilder
  - ✓”Bläddra” i böckerna
- Under föreläsningen:
  - ✓
  - ✓ VAR AKTIV (är du trött – stanna hemma och sov)
- Efter föreläsningarna:
  - ✓Bearbeta (går-)dagens material
  - ✓Använd simulatorerna, hemma och i skolan
- En gång i veckan:
  - ✓Repetera förra veckans mål
  - ✓Jobba med veckans mål
  - ✓Studera kommande veckans mål

Kursupplägg

Föreläsning	**Fö**
Simulatorövningar	**S**
Laboration	**L**



Schema Vanliga veckor LP1 HT11

Fö: Föreläsningar / Demonstration
Sim: Handledd självverksamhet.
LAB: Laboration

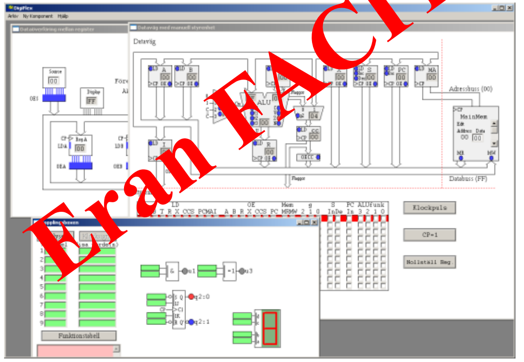
	Må	Ti	On	To	Fr
8				Sim B 08.00-09.45 ES61; LAB A 08.00-11.45 LAB 4225	LAB C 08.00-11.45 LAB 4225
10		Fö HA2	Fö HA2		Sim D 10.00-11.45 ES61
13	Fö HA2	Fö/Demo HA2		Sim C 13.15-15.00 ES61; LAB B 13.15-17.00 LAB 4225	LAB D 13.15-17.00 LAB 4225
15		Sim A 15.15-17.00 ES61			

SimStart Tis 30/8 Lab Start Tor 15/9

Simulator övningar

DigiFlex 7.5
Kopplingsboxen
Datavägar
Flex

Eterm 6.7
FLEX
HC12
I/O
Borrmaskin



Laborationsregler

- Laborationerna innehåller hemuppgifter. (Se respektive laborations-PM). Dessa uppgifter **skall** vara lösta och uppvisade **före** laborationstillfället.
- Vid laborationstillfället delas ut **extra uppgifter** som du självständigt skall lösa för att bli godkännd.
- En handledare skall bedöma arbetet efter laborationen.
- **Kom i tid** och hasta ej igenom uppgifterna.

Är laborant p g a sjukdom eller annan angelägen orsak förhindrad att delta vid ett laborationstillfälle skall detta omedelbart meddelas till laborationschefen.

Kurslitteratur

(KMP) Johnson, Larsson & Arebrink: Grundläggande digital- och datorteknik. Kompendium, Inst för datorteknik, CTH, 2007. Del 1. Digital teknik
(ARB1) Arbetsbok för DigiFlex. Ver 4, 2008.
[*(ARB2) Arbetsbok för HC12.]

(EXT) Kompletterande material CTH 2010
(INS1) Instruktionslista för FLEX-processorn.
(INS2) Instruktionslista för mikroprocessorn CPU12.
(LAB) Laborations-PM nr 1-4. Inst för datorteknik, CTH, 2010.
KIT

(SIM) Simulatorer för digitala kretsar, FLEX- processorn och HC12-processorn. Programvara för PC (Windows).

Fo1

Kursens mål:
- Använda en modern processor
- Konstruera en dator mha grindar och programmera denna

Veckans mål:
- Beskriva grindar och de verktyg som behövs under konstruktionsarbetet
- Hur kodas tal och tecken i datorn

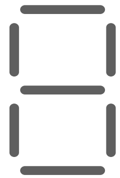
Dagens mål:
- Inledning till Digital o Datorteknik
- Binära tal
- Talomvandling
- Koder

Läs smart!
Lär dig mer!

DIGITAL- och DATORTEKNIK

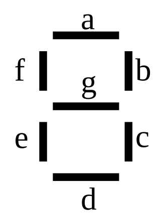
Vad är digitalteknik för något?

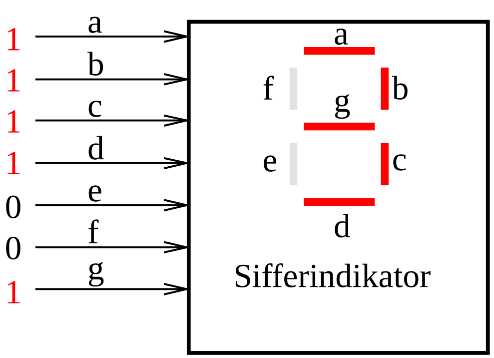
Digitalteknik = **Siffer**teknik



Sifferindikator

Sifferindikatorn har sju segment. Man kan utifrån välja vilka segment som skall synas och vilka som skall vara osynliga.

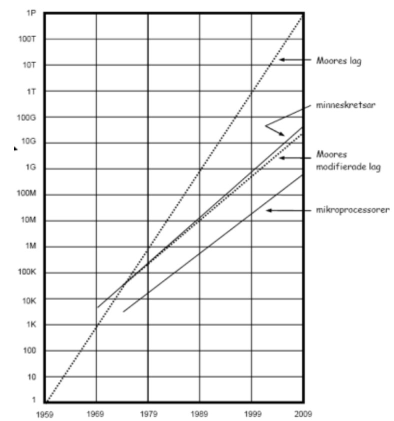




Ett **digitalt** system arbetar med **siffror**.

Mikroelektronikens utveckling

Antalet transistorer som ryms på en kiselbricka....



Vad Repr ettorna o nollorna

1001101000111...0101...10111000011

Vad representerar ettorna o nollorna?

- Binärtal
- Excess-kod
- Gray-kod
- NBCD
- ASCII-kod

Begrepp vid binär kodning

begrepp	*betydelse*	*exempel...*
bit/bitar	minsta informationsenhet, kan anta två värden	`0` eller `1`
bitsträng binärt ord	sekvens av bitar	`101100100001...`
kodord	$K_7\ K_6\ K_5\ K_4\ K_3\ K_2\ K_1\ K_0$ också ett binärt ord men med en fastställd kodning (betydelse)	`1000001` = "A" (ASCII) `1000001` = 65 (naturligt tal) `1000001` = -127 (heltal)
ordlängd	antal bitar i ordet	
nibble	ordlängden 4 bitar	`0101`
byte	ordlängden 8 bitar	`01011100`

Decimala – positionssystemet; binära tal

SEK 215:-

~~251:-~~
~~521:-~~
152:-

Decimala tal
Bas 10
10 siffror

$2 \cdot 10^2 \;+\; 1 \cdot 10^1 \;+\; 5 \cdot 10^0 \qquad 215_{10}$

Binära tal
Bas 2
2 siffror

$10111_2 \;=\; 1 \cdot 2^4 \;+\; 0 \cdot 2^3 \;+\; 1 \cdot 2^2 \;+\; 1 \cdot 2^1 \;+\; 1 \cdot 2^0$

$16 \qquad \cancel{8} \qquad 4 \qquad 2 \qquad 1 \;=\; 23_{10}$

Talsystem

Basen 2, 8, 10, 16

S2.4

$\beta = 2$ binärt	$\beta = 8$ oktalt	$\beta = 10$ decimalt	$\beta = 16$ 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

Omvandling av ett tal N med basen 10 till basen 2.

$(N)_{10} \rightarrow (N)_2$

Talet $(N)_{10}$ delas upp i heltalsdel $(N_H)_{10}$ och bråktalsdel $(N_B)_{10}$.

$(N)_{10} = (N_H . N_B)_{10}$

Heltalsdelen och bråktalsdelen behandlas sedan var för sig.

Exempel
211.678_{10} till binärt

Heltalsdelen:

Heltalsdelen skall skrivas med siffor i basen 2 som:

$$(N_H)_2 = d_{n-1}\, d_{n-2} \dots d_0$$

Man tänker sig heltalsdelen av talet $(N_H)_{10}$ skrivet med nya basen 2.

$$(N_H)_{10} = d_{n-1}2^{n-1} + d_{n-2}2^{n-2} + \dots + d_1 2^1 + d_0 2^0$$

Dividera talet med basen 2.

$$(N_H)_{10} / 2 = \underbrace{d_{n-1}2^{n-2} + d_{n-2}2^{n-3} + \dots + d_1 2^0}_{\text{heltal}} + \underbrace{d_0 2^{-1}}_{\text{bråktal}}$$

Efter divisionen "dyker" d_0 upp som bråktalsdel (rest).
Fortsatt division av det nya heltalet med basen 2 ger siffrorna d_1, d_2, d_3, ... i talet, fast nu i basen 2.

Bråktalsdelen:

Bråktalsdelen skall skrivas med siffor i basen 2 som

$$(N_B)_2 = .\, d_{-1}\, d_{-2} \dots d_{-m}$$

Man tänker sig bråktalsdelen av talet $(N_B)_{10}$ skrivet i den nya basen 2.

$$(N_B)_{10} = d_{-1}2^{-1} + d_{-2}2^{-2} + \dots + d_{-m}2^{-m}$$

Multiplicera talet med basen 2.

$$2(N_B)_{10} = \underbrace{d_{-1}2^0}_{\text{heltal}} + \underbrace{d_{-2}2^{-1} + \dots + d_{-m}2^{-m+1}}_{\text{bråktal}}$$

Efter multiplikationen "dyker" d_{-1} upp som heltalsdel.
Fortsatt multiplikation av den nya bråktalsdelen med basen 2 ger siffrorna d_{-2}, d_{-3}, ... i talet, fast nu i basen 2.
Till sist sätts talet $(N)_2$ samman av heltalsdelen och bråktalsdelen.

$$(N)_2 = (N_H . N_B)_2 = (d_{n-1}\, d_{n-2} \dots d_0 \,.\, d_{-1}\, d_{-2} \dots d_{-m})_2$$

Heltalsdelen:

$(211)_{10}$ omvandlas till bas 2 genom successiva divisioner med 2.

	kvot		**rest**	
211/2 =	105	+	½	$d_0 = 1$
105/2 =	52	+	½	$d_1 = 1$
52/2 =	26	+	0	$d_2 = 0$
26/2 =	13	+	0	$d_3 = 0$
13/2 =	6	+	½	$d_4 = 1$
6/2 =	3	+	0	$d_5 = 0$
3/2 =	1	+	½	$d_6 = 1$
1/2 =	0 Sluta	+	½	$d_7 = 1$

som ger $(211)_{10} = (11010011)_2$

Bråktalsdelen:

Talet $(0,678)_{10}$ omvandlas till bas 2 med successiva multiplikationer

S2.12
Ext2

	heltal		bråk	
$2 \cdot 0,678 =$	1	+	0,356	$d_{-1} = 1$
$2 \cdot 0,356 =$	0	+	0,712	$d_{-2} = 0$
$2 \cdot 0,712 =$	1	+	0,424	$d_{-3} = 1$
$2 \cdot 0,424 =$	0	+	0,848	$d_{-4} = 0$
$2 \cdot 0,848 =$	1	+	0,696	$d_{-5} = 1$
$2 \cdot 0,696 =$	1	+	0,392	$d_{-6} = 1$
			:	:

Sluta ?????

som ger $(0,678)_{10} = (0.101011.....)_2$

Fo1

Kursens mål:

- Använda en modern processor
- Konstruera en dator mha grindar och programmera denna

Veckans mål:

- Beskriva grindar och de verktyg som behövs under konstruktionsarbetet
- Hur kodas tal och tecken i datorn

Dagens mål:

- Inledning till Digital o Datorteknik
- Binära tal
- Talomvandling
- **Koder**

Läs smart! Lär dig mer!

GRAY-KOD

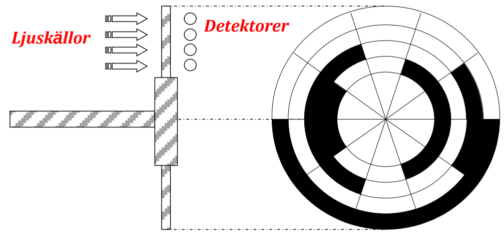
Tabell 2.2. Graykoder.

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

Gray kod

Kodskiva – vanlig komponent i olika typer av vinkelgivare.



I "övergångarna": Koder ändrar sig endast i en bit.
Förhindrar tillfälliga felavläsningar.

Tabell 2.1. Excess-2^{n-1} kodning vid n = 4.

S2.15

Excess-kod

Ex avkoda:

1 1 0 0_2 Excess-2^{n-1}

n=4

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

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

S2.18

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

Alfanumeriska tecken → ASCII

American Standard Code for Information Interchange

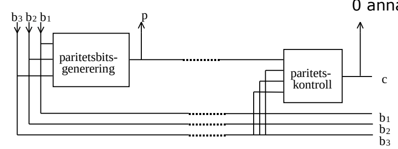
Typiskt användningsområde: Tangentbord

Felupptäckande kod - Paritetsbitar

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

Extra bit läggs till så att ordet alltid innehåller jämnt antal ettor (jämn paritet)

Kontrollbit:
1 om udda antal ettor
0 annars

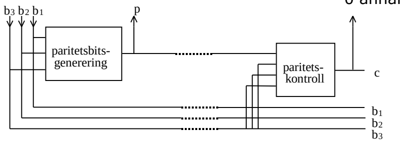


Exempel: Jämn paritetsmetod använd i samband med 3-bitars kod.

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

Extra bit läggs till så att ordet alltid innehåller udda antal ettor (udda paritet)

Kontrollbit:
1 om jämnt antal ettor
0 annars



Exempel: Udda paritetsmetod använd i samband med 3-bitars kod.

33

Sammanfattning

Något bitmönster: 101100

Kan anges på HEX-form: 2C

Kan representera ett binärtal: 101100_2

Kan anges på HEX-form: $2C_{16}$

Motsvarande decimala tal:

$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 32+8+4=44_{10}$

Motsvarande decimala tal:

$2 \cdot 16^1 + C \cdot 16^0 = 2 \cdot 16 + 12 \cdot 1 = 44_{10}$

NBCD-talen: Binär/Hex-talen [0,9]

Vad Repr ettorna o nollorna

Binärtal	*0001, 0010, 0011, 0100, etc*
Excess-kod	*Förskjuten nollpunkt; plus o minus*
Gray-kod	*Enhetsavstånd; en bit ändrar sig*
NBCD	*Siffrorna 0,9; (A,F; 1010,1111, etc används ej)*
ASCII-kod	*Kod för text (tangentbord)*

Kursens mål:

Digital teknik

Dator teknik

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

… och därmed kunna använda en modern µ-processor / µ-controller

I DAG

Fö2

Veckans mål:

- Beskriva grindar och de verktyg som behövs under konstruktionsarbetet av datorn
- Hur kodas tal och tecken i datorn

Dagens mål:

- Switchnätsalgebra
- Boolesk algebra
- Funktionstabell
- Binär evaluering
- Grindar

Läs smart!
Lär dig mer!

Verktyg

- **Boolesk algebra**
- **Funktionstabell**
- **Karnaughdiagram**

3.2 Boolesk algebra

S3.7

Operatorer: +, •, ', **Data:** 0, 1

Räkneregler:

Eller	Och	Icke
0 + 0 = 0	0·0 = 0	0' = 1
0 + 1 = 1	0·1 = 0	1' = 0
1 + 0 = 1	1·0 = 0	
1 + 1 = 1	1·1 = 1	

Satser inom Boolesk algebra.

S3.8

1. Kommutativa lagarna

$x + y = y + x$

$x \cdot y = y \cdot x$

2. Distributiva lagarna

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

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

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. $x + x = x$, $x \cdot x = x$

7. Associativa lagarna

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

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

8. De Morgans lagar

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

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

9. $(x')' = x$

Exempel 3.6
Förenkla
f = X + XY

S3.9

f = X + XY

f = X · 1 + XY sats 3

f = X · (1 + Y) sats 2

f = X · 1 sats 5

f = X sats 3

Funktionstabell

S3.8

Satserna i Boolesk algebra kan enkelt bevisas med hjälp av binär evaluering och utnyttjande av räknereglerna för en Boolesk algebra.

Enklast görs detta i tabellform i en s k funktionstabell.

Visa att: (x + y)' = x' · y'

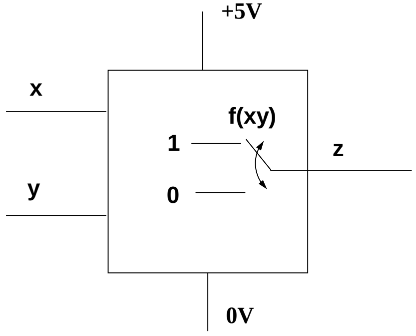
(Sats 8 De Morgans)

Sätt upp alla kombinationer av variablerna !!!!

Logikkrets.

S3.18
S1.14

Grind
ELLER (OR)
OCH (AND)
INVERTERARE (ICKE, NOT)
NOR
NAND

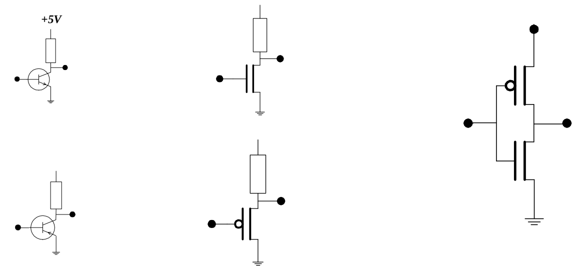


Några olika teknologier...

TTL (Transistor-Transistor-Logic)

MOS (Metal Oxide Silicon)

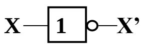
CMOS (Complementary MOS)



Mer om detta kommer i kursen Digitalteknik FK

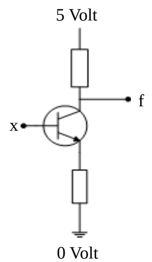
S3.18
Arb App E

Funktionstabeller för grundläggande grindtyper



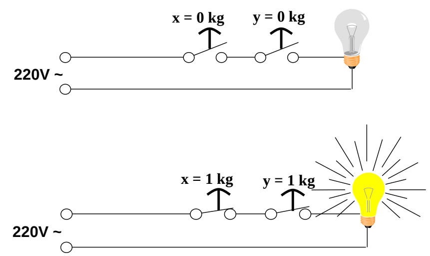
Inverterare (NOT)

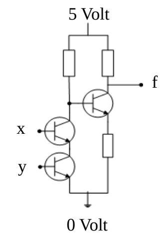
X	f
0	1
1	0



AND - GRIND

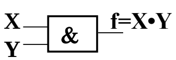
x	y	f
0	0	0
0	1	0
1	0	0
1	1	1





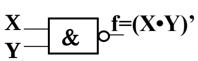
Funktionstabeller för grundläggande grindtyper

S3.19
Arb App E



AND (OCH)

X	Y	f
0	0	0
0	1	0
1	0	0
1	1	1

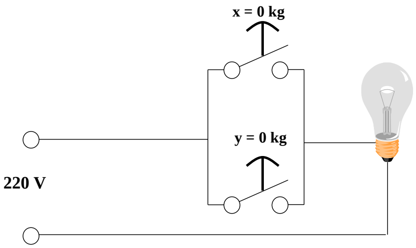


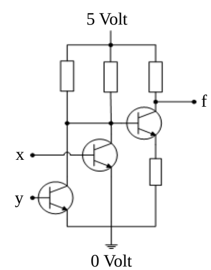
NAND

X	Y	f
0	0	1
0	1	1
1	0	1
1	1	0

OR - GRIND

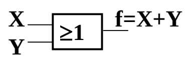
x	y	f
0	0	0
0	1	1
1	0	1
1	1	1





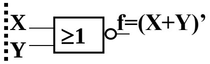
Funktionstabeller för grundläggande grindtyper

S3.19
Arb App E



OR (ELLER)

X	Y	f
0	0	0
0	1	1
1	0	1
1	1	1



NOR

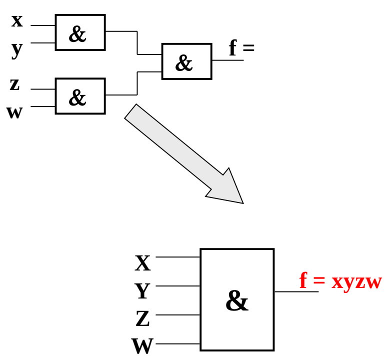
X	Y	f
0	0	1
0	1	0
1	0	0
1	1	0

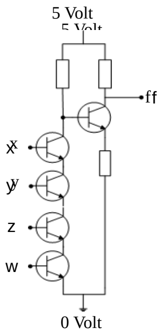
Logikkretssymboler för grundläggande logikoperationer.

S3.18
S1.14

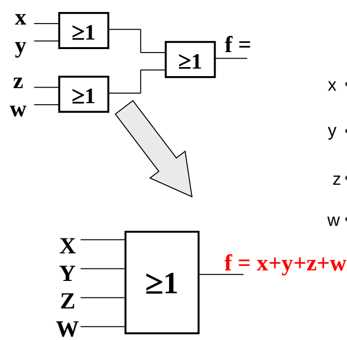
Funktion	*Grind*	*Grafisk symbol*
z = x+y	ELLER (OR)	x, y → ≥ 1 → z
z = x·y	OCH (AND)	x, y → & → z
z = x'	INVERTERARE (ICKE, NOT)	x → 1 ○ → z
z = (x+y)'	NOR	x, y → ≥ 1 ○ → z
z = (x·y)'	NAND	x, y → & ○ → z

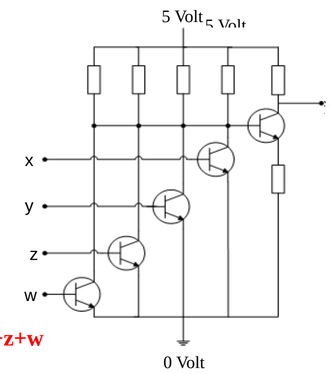
Utökningar av grindar





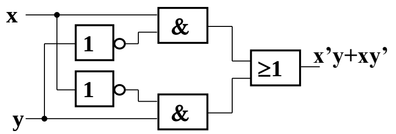
Utökningar av grindar





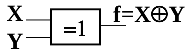
S3.23

XOR, (Exclusive-OR)



Speciell operationssymbol ⊕

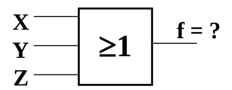
Så: $x \oplus y = x'y + xy'$

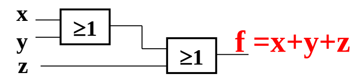


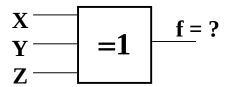
XOR

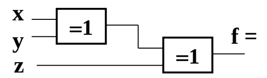
X	Y	f
0	0	0
0	1	1
1	0	1
1	1	0

Utökningar av grindar









x y z		f
0 0 0		
0 0 1		
0 1 0		
0 1 1		
1 0 0		
1 0 1		
1 1 0		
1 1 1		

I DAG

Fö3

Veckans mål:

- Beskriva grindar och de verktyg som behövs under konstruktionsarbetet av datorn
- Hur kodas tal och tecken i datorn

Dagens mål:

- Booleska uttryck och funktioner
- Disjunktiv / Konjunktiv form
- Normal / Minimal form
- Mintermer / Maxtermer
- Karnaughdiagram
- Minimering av grindnät
- NAND / NOR logik

Läs smart!
Lär dig mer!

- **Begrepp**
  - **Booleska uttryck och funktioner**
  - **Disjunktiv / Konjunktiv form**
  - **Normal / Minimal form**
  - **Mintermer / Maxtermer**
  - **Karnaughdiagram**
  - **Minimering av grindnät**

Exempel Kalle: Realisera

$f(x,y,z) = xy+yz+x'z$

med grindar

Booleska uttryck och Booleska funktioner s3.14-17

Booleska uttryck som är ekvivalenta repr. samma **Booleska funktion**:

f(x,y) = (x+y)' och g(x,y) = x'·y' utgör olika uttryck för samma funktion (ty (x+y)' = (x'·y') enligt sats 8). (jfr vanlig algebra)

Ett uttryck på **disjunktiv form** är en summa av termer, där varje term är en Boolesk produkt av variabler (med eller utan prim):
T ex f(x,y,w,z) = yz + wz + x'yw'

Ett uttryck på **konjunktiv form** är en produkt av faktorer, där varje faktor är en Boolesk summa av variabler (med eller utan prim):
T ex g(x,y,w,z) = (y + w)(w' + z)(x' + z)

Exempel (Kalle forts)

Visa hur det Booleska uttrycket

f(x,y,z) = xy+yz+x'z

1) kan förenklas (minimeras) till

disjunktiv minimal form: (*Summa av Produkter*)

2) kan förenklas (minimeras) till

konjunktiv minimal form: (*Produkt av Summor*)

Exempel (Kalle forts)

Ext 3
Kmp s 3.9

Visa hur f(x,y,z) = xy+yz+x'z kan förenklas till disjunktiv (minimal) form: (*Summa av Produkter*)

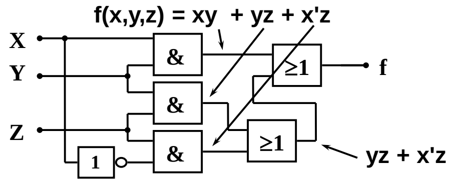
f(x,y,z)	=	xy + yz + x'z	
	=	xy + yz1 + x'z	sats 3
	=	xy + yz(x + x') + x'z	sats 4
	=	xy + yzx + yzx' + x'z	sats 2
	=	xy + xyz + x'z + x'yz	sats 1
	=	xy1 + xyz + x'z1 + x'zy	sats 3
	=	xy(1 + z) + x'z(1 + y)	sats 2
	=	xy1 + x'z1	sats 5
	=	xy + x'z	sats 3

Exempel (Kalle forts)

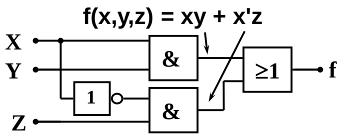
Visa hur f(x,y,z) = xy+yz+x'z kan förenklas till konjunktiv (minimal) form: (*Produkt av summor*)

f(x,y,z)	=	xy + yz + x'z	
	=	xy + yz + xx' + x'z	sats 4
	=	y(x + z) + x'(x + z)	sats 2
	=	(x + z) (y + x')	sats 2

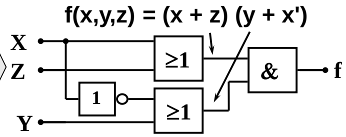
Grindnät för exempel Kalle



Disjunktiv (minimal) form
(*Summa av Produkter*)



Konjunktiv (minimal) form
(*Produkt av summor*)



Disjunktiv normal form och konjunktiv normal form

Vårat gamla exempel Kalle: f(x,y,z) = x y + y z + x'z

Rita funktionstabell
(3 variabler x,y,z => 2^3 = 8 rader)

Disjunktiv normal form och konjunktiv normal form

Vårat gamla exempel Kalle: f(x,y,z) = x y + y z + x'z
Rita funktionstabell (3 variabler => 2^3 = 8 rader)

	x y z	xy	yz	x'z	f
0)	0 0 0	0	0	0	0
1)	0 0 1	0	0	1	1
2)	0 1 0	0	0	0	0
3)	0 1 1	0	1	1	1
4)	1 0 0	0	0	0	0
5)	1 0 1	0	0	0	0
6)	1 1 0	1	0	0	1
7)	1 1 1	1	1	0	1

	x y z	f
0)	0 0 0	0
1)	0 0 1	1
2)	0 1 0	0
3)	0 1 1	1
4)	1 0 0	0
5)	1 0 1	0
6)	1 1 0	1
7)	1 1 1	1

Invariabelkombinationen (x, y, z) i varje rad i funktionstabellen är unik.

Ta fram en unik produkt av invariabler som ger 1 · 1 · 1 = 1. (*T.ex ger rad 6 x y z'*)

Ta fram produkter för samtliga rader där *funktionsvärdet är 1* och adderar dessa. Summan blir ett uttryck för den Booleska funktionen f = x'y'z + x'y z + x y z' + x y z

Varje term är unik och innehåller samtliga invariabler.
Termerna kallas mintermer och funktionen sägs vara skriven på disjunktiv normal form.

Ext 3
S3.16

	x	y	z	f
0)	0	0	0	0
1)	0	0	1	1
2)	0	1	0	0
3)	0	1	1	1
4)	1	0	0	0
5)	1	0	1	0
6)	1	1	0	1
7)	1	1	1	1

Invariabelkombinationen (x, y, z) i varje rad i funktionstabellen är unik.

Ta fram en unik summa av invariabler som ger 0+0+0=0. (*T.ex ger rad 4 x'+y +z)*

Ta fram summor för samtliga rader där funktionsvärdet är 0 och multiplicera dessa. Produkten blir ett uttryck för den Booleska funktionen

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

Varje faktor är unik och innehåller samtliga invariabler. Faktorerna kallas maxtermer och funktionen sägs vara skriven på konjunktiv normal form.

I DAG

Fö3

Veckans mål:

- Beskriva grindar och de verktyg som behövs under konstruktionsarbetet av datorn
- Hur kodas tal och tecken i datorn

Dagens mål:

- Booleska uttryck och funktioner
- Disjunktiv / Konjunktiv form
- Normal / Minimal form
- Mintermer / Maxtermer
- **Karnaughdiagram (för att minimera)**
- Minimering av grindnät
- NAND / NOR logik

Läs smart!
Lär dig mer!

Funktionstabell

	x	y	z	f (x, y, z)
0)	0	0	0	0
1)	0	0	1	1
2)	0	1	0	0
3)	0	1	1	1
4)	1	0	0	0
5)	1	0	1	0
6)	1	1	0	1
7)	1	1	1	1

Gray-kodat

Alternativ uppställning S3.25-27

	x	y	z	f (x, y, z)
0)	0	0	0	0
1)	0	0	1	1
3)	0	1	1	1
2)	0	1	0	0
6)	1	1	0	1
7)	1	1	1	1
5)	1	0	1	0
4)	1	0	0	0

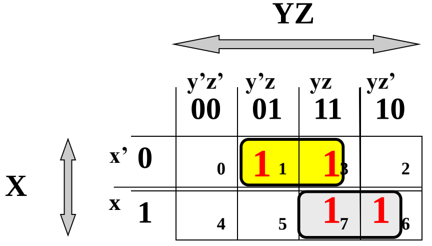
1) och 3) $x'y'z + x'yz = x'z(y' + y) = x'z$

3) och 7) $x'yz + xyz = yz(x' + xy) = yz$

6) och 7) $xyz' + xyz = xy(z' + z) = xy$

$f = xy + yz + x'z$

Karnaughdiagram



X är noll
Y är don't care
Z är ett
X'Z

X är ett
Y är ett
Z är don't care
XY

$f_{min} = X'Z + XY$

S3.30

f =

wx \ yz	00	01	11	10
00				
01	1	1	1	1
11				
10				

f =

wx \ yz	00	01	11	10
00				
01				
11	1	1		
10	1	1		

f =

wx \ yz	00	01	11	10
00		1	1	
01				
11				
10		1	1	

figurerna 3.27 och 3.28

f =

wx \ yz	00	01	11	10
00	1			1
01				
11				
10	1			1

f =

wx \ yz	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11				
10				

f =

wx \ yz	00	01	11	10
00	1			1
01	1			1
11	1			1
10	1			1

Sammanfattning

- •NORMAL form ⇒ Funktionstabell
- •MINIMAL form ⇒ Kranaughdiagram

•DISJUNKTIV (*normal / minimal*) form

- •Σ av Prod Ex: (x'y)+(xw)+(xyw)
- •Ettor
- •Mintermer: (1• 1• 1) = 1
- •NAND / NAND - logik

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

I DAG

Fö3

Veckans mål:

- Beskriva grindar och de verktyg som behövs under konstruktionsarbetet av datorn
- Hur kodas tal och tecken i datorn

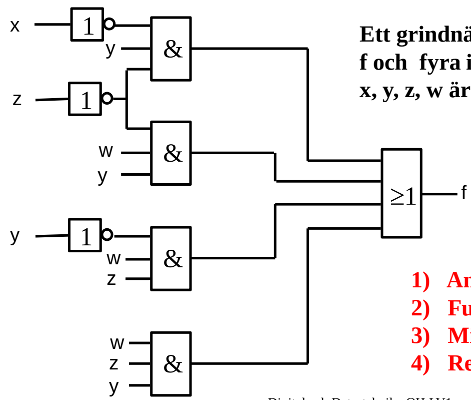
Dagens mål:

- Booleska uttryck och funktioner
- Disjunktiv / Konjunktiv form
- Normal / Minimal form
- Mintermer / Maxtermer
- Karnaughdiagram
- **Minimering av grindnät**
- NAND / NOR logik

Läs smart!
Lär dig mer!

Praktikfall, minimering av grindnät

Ext4



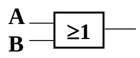
Ett grindnät med utsignalen f och fyra insignalerna x, y, z, w är givet.

Kan man konstruera ett "mindre" nät?

1) Analys
2) Funktionstabell
3) Minimering (Karnaugh)
4) Realicering

NAND-logik

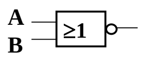
Om vi har A+B

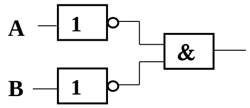


Hur realicera med NAND?

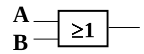
De Morgan:

(A+B)’ = A’B’

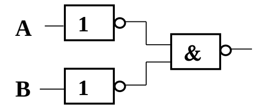




NAND-logik



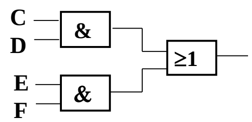
Samma sak

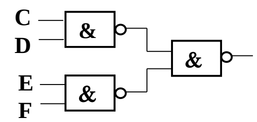


---

om nu A = f(C,D) och B = f(E,F)

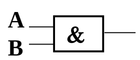
Disjunktiv form





NOR-logik

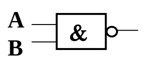
Om vi har A•B

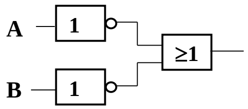


Hur realicera med NOR?

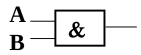
De Morgan:

(AB)’ = A’ + B’

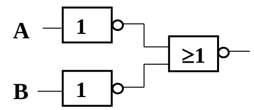




NOR-logik



Samma sak



---

om nu A = f(C,D) och B = f(E,F)

Konjunktiv form