

F8: Logiska komponenter

- Innehåll:
 - Avkodare
 - Digitala koder
 - 2-4 avkodare
 - 7-segments display
 - Kodare
 - Multiplexer
 - Demultiplexer

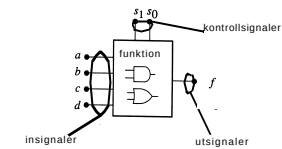
1 (2 2)

Introduktion

• Logiska komponenter

Det är nödvändigt att skapa mer komplexa byggblock än de grundläggande grindarna (*and*, *or*, *not*) som kan återanvändas som färdiga byggblock.

• Generell modell:



$$\begin{aligned}
 f &= f_0(a, b, c, d) & \text{if } s_1s_0 &= 00 \\
 f &= f_1(a, b, c, d) & \text{if } s_1s_0 &= 01 \\
 f &= f_2(a, b, c, d) & \text{if } s_1s_0 &= 10 \\
 f &= f_3(a, b, c, d) & \text{if } s_1s_0 &= 11
 \end{aligned}$$

Dataflöde:

- in- och utgångar (a,b,c,d resp. f)

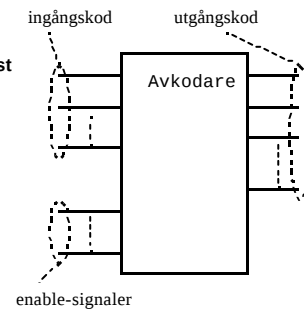
Kontrollsignaler:

- styr dataflödet (s_0, s_1)

2 (2 2)

Avkodare

- Krets med flera ingångar och flera utgångar
- Antalet ingångar (n) är mindre än antalet utgångar (m)
- Konverterar en ingångskod till en utgångskod
- 1 - till - 1 mappning
 - d.v.s för varje ingångskod finns det endast en utgångskod
- Exempel på ingångskoder
 - binär kod
 - Gray kod
 - BCD kod
 - vilken-som-helst kod



3 (2 2)

Koder

Decimal digit	BCD (8421)	2421	Excess-3	Biquinary	1-out-of-10
0	0000	0000	0011	0100001	100000000
1	0001	0001	0100	0100010	010000000
2	0010	0010	0101	0100100	001000000
3	0011	0011	0110	0101000	000100000
4	0100	0100	0111	0110000	000010000
5	0101	1011	1000	1000001	000001000
6	0110	1100	1001	1000010	000000100
7	0111	1101	1010	1000100	000000010
8	1000	1110	1011	1001000	000000001
9	1001	1111	1100	1010000	000000000
Unused code words					
	1010	0101	0000	0000000	000000000
	1011	0110	0001	0000001	000000001
	1100	0111	0010	0000010	000000010
	1101	1000	1101	0000011	000000011
	1110	1001	1110	0000101	000000011
	1111	1010	1111	...	...

4 (2 2)

Gray koder

Decimal Number	Binary Code	Gray Code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

- En n-bits Gray-kod konstrueras som följer genom upprepning:
 - En 1-bits Gray-kod har två kodord, 0 och 1.
 - De 2^n första kodorden av en n+1 bitars Gray-kod är lika med kodord för en n-bitars Gray-kod skriven i samma ordning med tillägg av en 0:a längst fram
 - De 2^n sista kodorden av en n+1 bitars Gray-kod är lika med kodord för en n-bitars Gray-kod skriven i omvänd ordning med tillägg av en 1:a längst fram

ASCII kod

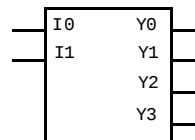
		$b_7b_6b_5b_4$ (column)							
$b_3b_2b_1b_0$	Row (hex)	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000	0	NUL	DLE	SP	0	@	P	'	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS	,	<	L	\	l	
1101	D	CR	GS	=	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O	_	o	DEL

2-4 Binär avkodare

- Generellt: n-till- 2^n avkodare : n ingångar och 2^n utgångar
- Ingångskod: binär
- utgångskod: 1-av- 2^n . En utgång för varje ingångskod
- Exempel: n=2, 2-till-4 avkodare

Ingångar		Utgångar			
I1	I0	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

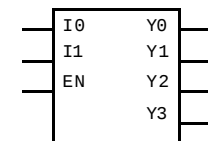
2-to-4 avkodare



2-4 Binär avkodare med "enable"

Ingångar			Utgångar			
EN	I1	I0	Y3	Y2	Y1	Y0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

2-to-4 avkodare

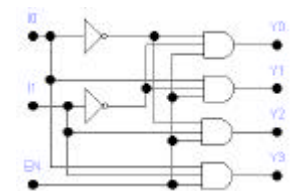


$$Y0 = I0' \cdot I1' \cdot EN$$

$$Y1 = I0 \cdot I1' \cdot EN$$

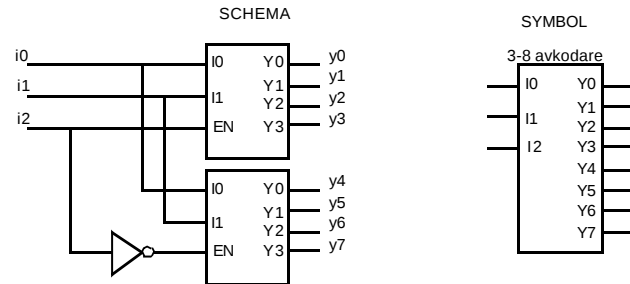
$$Y2 = I0' \cdot I1 \cdot EN$$

$$Y3 = I0 \cdot I1 \cdot EN$$



Kaskadkoppling av avkodare

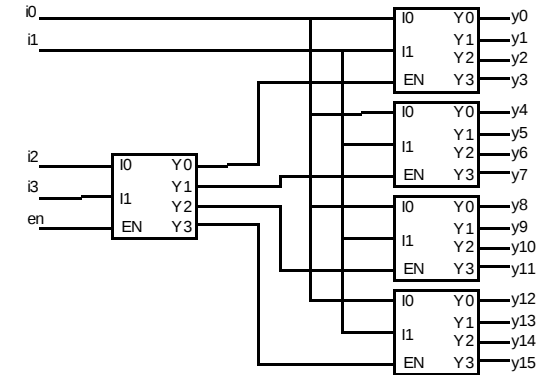
- Konstruera en 3-8 avkodare med 2-4 avkodare



9 (22)

Kaskadkoppling av avkodare

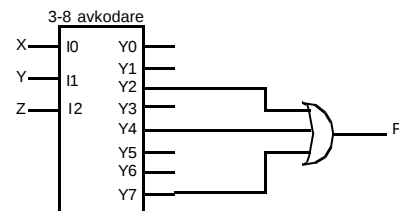
- Konstruera en 4-16 avkodare med 2-4 avkodare



10 (22)

Användning av binär-avkodare

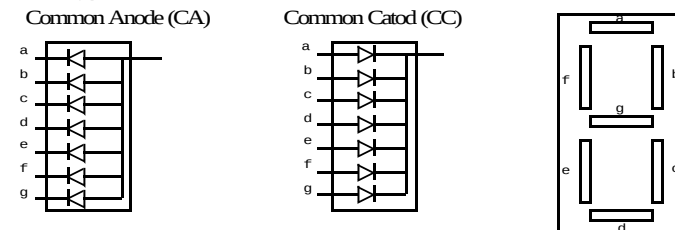
- Skapa summa-av-produkt uttryck
 - Den binära avkodaren genererar alla mintermer för en funktion med n-variabler
 - Summa-av-produkten för den logiska funktionen fås genom att göra logisk ELLER med funktionens mintermer.
 - Exempel: $F = \sum_{x,y,z} (2, 4, 7)$



11 (22)

Sju-segments-display

- Visar decimala siffror och en del alfabetiska tecken
- LED (Light Emitting Diode) eller LCD (Liquid Crystal Display)
- LED-typ:



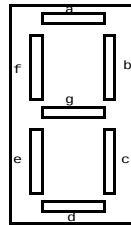
- CA: kräver ingångar som är aktivt låga (en drivare med aktivt låga utgångar)
- CC: kräver utgångar som är aktivt höga (aktivt höga utgångar)

12 (22)

Sju-segment avkodare/drivare

- Ingångskod: BCD
Utgångskod: sju-segment-kod
- Sanningstabell för en aktiv-hög avkodare

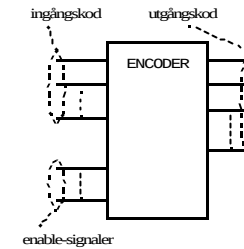
Ingångar				Utgångar						
D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	0	0	1	1	0	1	1	0	1
0	0	0	1	1	1	1	1	0	0	1
0	1	1	0	0	1	1	0	0	1	1
0	1	1	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1



13 (22)

Kodare (encoder)

- Krets med många ingångar och många utgångar
- Utför den motsatta funktionen av en avkodare
- Antal utgångar (m) är färre än antalet ingångar (n)
- Konverterar ett ingående kodord till ett utgående kodord

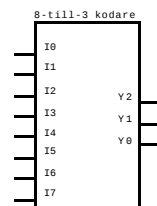


14 (22)

Binär kodare

- 2^n -till-n kodare : 2^n ingångar och n utgångar
- Ingångskod: 1-av- 2^n
- Utgångskod: Binär kod
- Exempel: n=3, m=8, 8-till-3 kodare

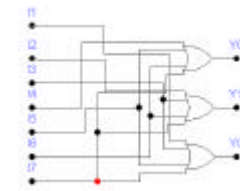
I0	I1	I2	I3	I4	I5	I6	I7	Y0	Y1	Y2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



15 (22)

Implementering av 8-till-3 kodare

- Från sanningstabellen fås ekvationerna
 - $Y2 = I1 + I3 + I5 + I7$
 - $Y1 = I2 + I3 + I6 + I7$
 - $Y0 = I4 + I5 + I6 + I7$
- Begränsningar
 - I0 har ingen påverkan på utgångarna
 - Endast en ingång får vara aktiv samtidigt

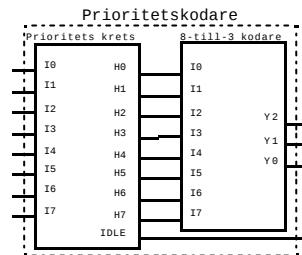


16 (22)

Prioritetskodare (priority encoder)

- Ingångarna har en inbördes prioritet
- När mer än en ingång är aktiv så genereras en kod för den ingång med högst prioritet
- Prioritetskodare

$H7 = I7$
 $H6 = I6 \cdot H7'$
 $H5 = I5 \cdot I6' \cdot H7'$
 $H4 = I4 \cdot I5' \cdot I6' \cdot H7'$
 $H3 = I3 \cdot I4' \cdot I5' \cdot I6' \cdot H7'$
 $H2 = I2 \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot H7'$
 $H1 = I1 \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot H7'$
 $H0 = I0 \cdot I1' \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot H7'$
 $IDLE = I0' \cdot I1' \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot H7'$

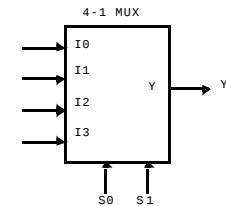


17 (22)

Multiplexer

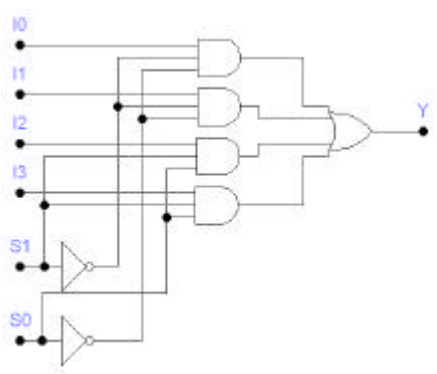
- Multiplexing: överföra ett stort antal signaler över en litet antal kanaler eller ledningar
- Digital multiplexer (MUX): väljer en av många ingångssignaler och dirigerar den till en enda utgång
- kontrollsignaler väljer en specifik ingång
- n stycket kontrollsignaler kan välja en av 2^n insignaler
- Exempel: 4-till-1 multiplexer:

S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3



18 (22)

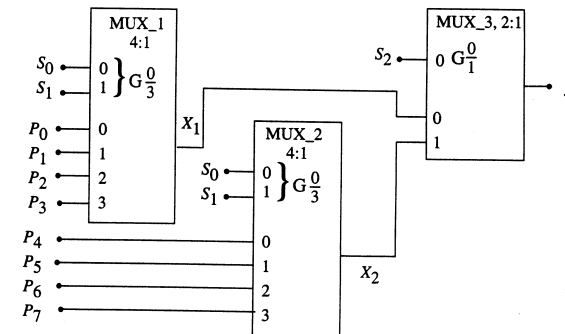
Logikschema för en 4-till-1 multiplexer



19 (22)

Konstruktion av 8-1 multiplexer

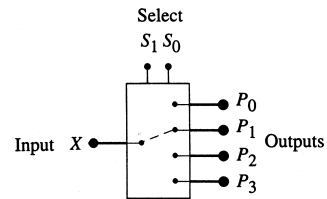
Datasignaler in: $P_7 - P_0$
 Datasignal ut: f
 Kontrollsignaler: $S_2 - S_0$



20 (22)

Demultiplexer

- En demultiplexer (DMUX) har den motsatta funktionen till en multiplexer
- En digital demultiplexer tar emot indata från en enda ingång och dirigerar den till en-av-många utgångar enligt kontrollsignalens värde



S_1	S_0	X to
0	0	P_0
0	1	P_1
1	0	P_2
1	1	P_3

Demultiplexer

- MUX/DMUX används i seriell dataöverföring

