Dugga 2, 2010-04-12

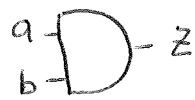
Uppgift 1:

A) AND-grind, OCH-grind. Släpper endast igenom signal om alla ingångar är satta till 1.

Sanningstabell:

a	b	z
0	0	0
0	1	0
1	0	0
1	1	1

om →

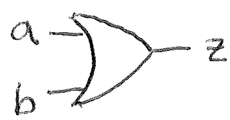


B) OR-grind, ELLER-grind. Släpper igenom signal om minst en ingång är satt till 1.

Sanningstabell:

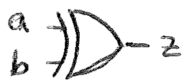
a	b	z
0	0	0
0	1	1
1	0	1
1	1	1

om →



C) Symboler:

Amerikansk



Europeisk

Booleskt uttryck: $a \oplus b = z$

eller $z = a \oplus b$

Uppgift 2:

0100111110100010_2

A) Till decimalt:

2048_{10} 512_{10} 128_{10} 32_{10} 2_{10}

0100111110100010_2

16384_{10} 1024_{10} 256_{10}

$16384 + 2048 + 1024 + 512 + 256 + 128 + 32 + 2 = \underline{\underline{20386_{10}}}$

B) Till oktalt:

Okej att lägga till nollor här

Inte här dock!

$00\,0|100|111|110|100|010$

0_8 4_8 7_8 6_8 4_8 2_8

$\Rightarrow \underline{\underline{47642_8}}$

C) Till hexadecimalt:

$0100|1111|1010|0010$

4_{16} F_{16} A_{16} 2_{16}

$\Rightarrow \underline{\underline{4FA2_{16}}}$

Uppgift 3:

A) $(\bar{a}+c)(a+c)=c$

Här kan man hävda att enligt lag T10' (Combining) så stämmer det, vilket är korrekt.
Vill man dock bevisa detta genom uträkning gör man på följande sätt:

$$(\bar{a}+c)(a+c)=\underbrace{\bar{a}\cdot a}_{=0}+\bar{a}\cdot c+c\cdot a+\underbrace{c\cdot c}_{=c} =$$

$$=\bar{a}\cdot c+c\cdot a+c = c\underbrace{(\bar{a}+a+1)}_{\substack{x \text{ eller } 1 \\ x \text{ OR } 1}} = c\cdot 1 = \underline{\underline{c}}$$

x eller 1 / x OR 1 } blir alltid 1

Sant!

B) Här finns det ingen enstaka lag att ta till, så här måste man räkna.

$$(a+b+c)(c+\bar{d}+a)=a\cdot c+a\cdot\bar{d}+\underbrace{a\cdot a}_{=a}+b\cdot c+b\cdot\bar{d}+$$

$$+b\cdot a+\underbrace{c\cdot c}_{=c}+c\cdot\bar{d}+c\cdot a =$$ [Flera termer som säger samma sak, som $a\cdot c$, kan reduceras till <u>en</u> term] =

$$=\underbrace{a\cdot c+a\cdot\bar{d}+a+b\cdot a}_{\text{bryt ut } a}+\underbrace{b\cdot c+c+c\cdot\bar{d}}_{\text{bryt ut } c}+b\cdot\bar{d} =$$

$$=a\underbrace{(c+\bar{d}+1+b\cdot a)}_{=1}+c\underbrace{(b+1+\bar{d})}_{=1}+b\cdot\bar{d} =$$

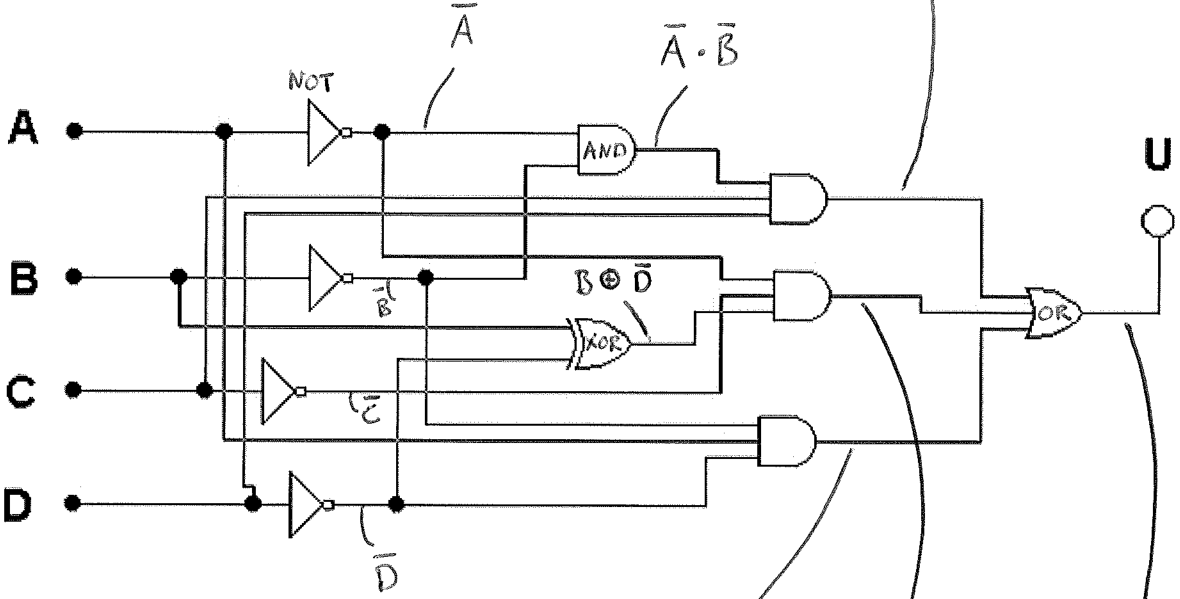
$= a+c+b\cdot\bar{d}$ som <u>inte</u> är detsamma som $a+c$

Uttrycket är alltså <u>falskt!</u>

Uppgift 4:

Enklast är det om man tar det steg för steg.

$(\bar{A}\cdot\bar{B})\cdot C\cdot D$



$A\cdot\bar{B}\cdot\bar{D}$

$\bar{A}\cdot\bar{C}\cdot(B\oplus\bar{D})$

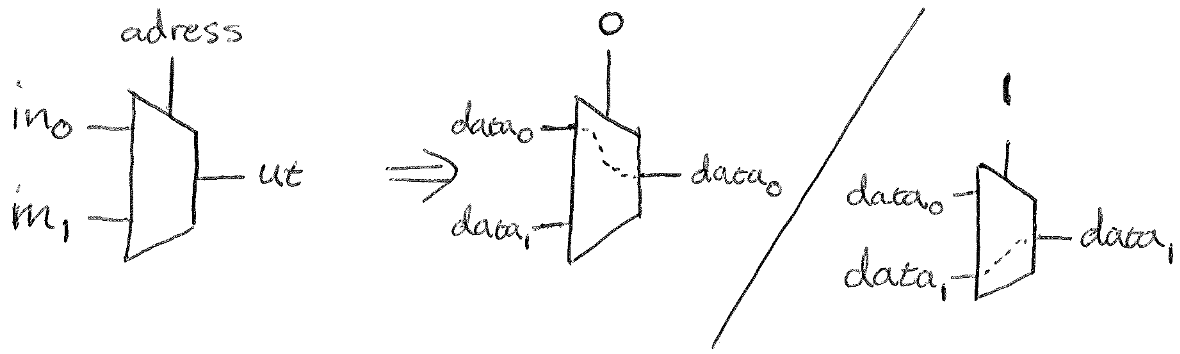
$$\left((\bar{A}\cdot\bar{B})\cdot C\cdot D\right)+\left(A\cdot\bar{B}\cdot\bar{D}\right)+\left(\bar{A}\cdot\bar{C}\cdot(B\oplus\bar{D})\right)$$

vilket är svaret.

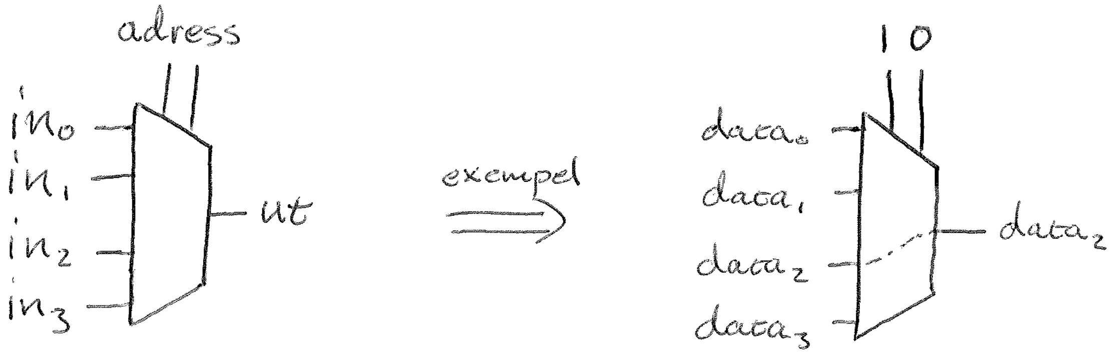
$$\Rightarrow U(A,B,C,D)=(\bar{A}\cdot\bar{B}\cdot C\cdot D)+(A\cdot\bar{B}\cdot\bar{D})+\left(\bar{A}\cdot\bar{C}\cdot(B\oplus\bar{D})\right)$$

Uppgift 5:

En multiplexer är en komponent där man bestämmer vilken av ingångarna som ska kopplas/skickas till utgången.
Detta bestämmer man genom en adressingång.
I det enklaste fallet har man en 2-till-1-mux:

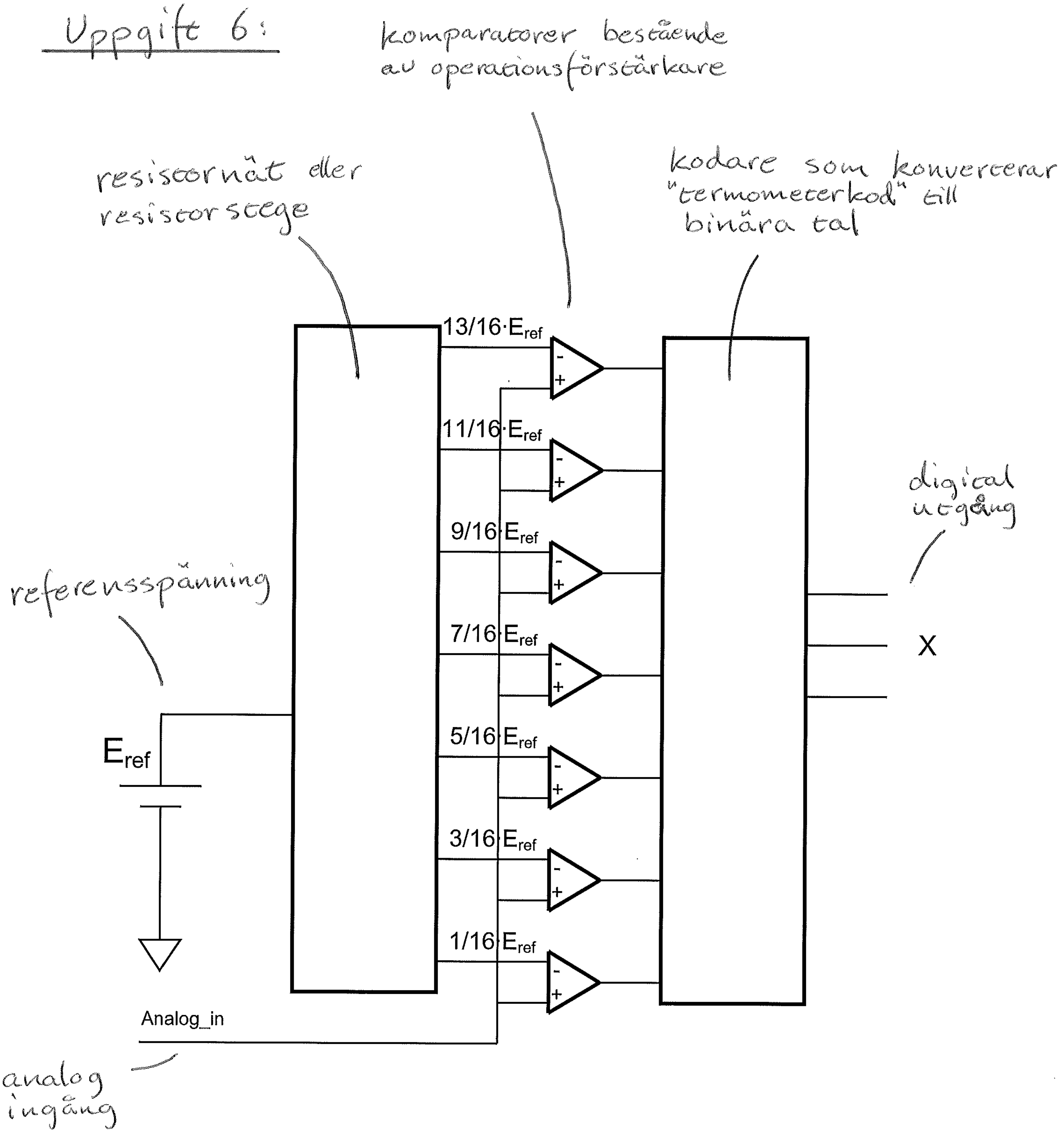


Beroende på adressen skickas datat (0 eller 1, ofta i en ström av data) från olika ingångar till utgången.
Ex. på 4-till-1:



Här behövs 2 adressbitar för att kunna representera 4 ingångar. 3 adressbitar till 8 ingångar osv.

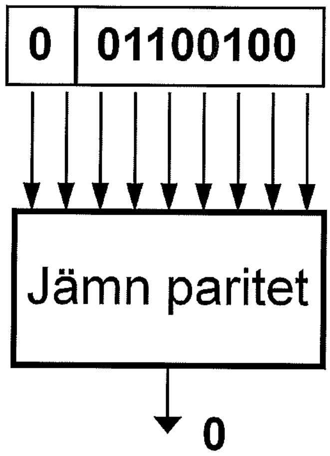
Uppgift 6:

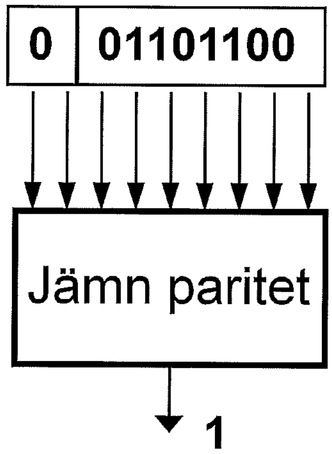


Resistornätet delar ner referensspänningen till $\frac{1}{16}E_{ref}$, $\frac{3}{16}E_{ref}$ osv. för att komparatorerna ska ha lämpliga spänningar (med linjärt mellanrum) att jämföra med. Kommer det exempelvis in en spänning på $\frac{4}{16}E_{ref} = \frac{1}{4}E_{ref}$ kommer de två nedersta komparatorerna ge ettor och övriga ge nollor. Kodaren kommer då att konvertera "talet 2" till binärkod, och ge $x = 010_2$.

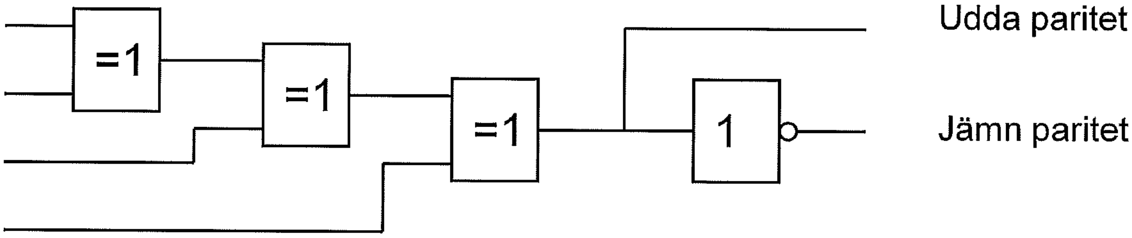
Uppgift 7:

En paritetskrets räknar antalet ettor i en bitström. I exemplet nedan har vi en 8-bitars dataström med en extra header-bit.
Jämn paritet → kretsen ger en etta om antalet ettor är jämnt.





En sådan krets kan implementeras med grindar (XOR och en NOT) på följande sätt:



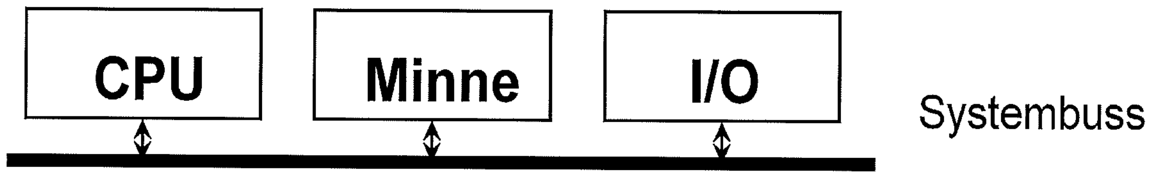
Implementationen är dock överkurs för denna uppgift, men man kan ju få en guldstjärna. :)

Vitsen med en paritetskrets är att man kan kolla pariteten innan sänder iväg informationen över en buss eller ~~ett~~ trådlöst nätverk, och sedan kolla om man har samma paritet när informationen kommer fram. Man kan då upptäcka om det blivit fel värde på en bit. Är det dock fel på ett jämnt antal bitar (t.ex. två) kommer inte paritetskodaren signalera att något är fel, så där ligger begränsningen.

Uppgift 8:

Den stora skillnaden ligger i att man har 2 bussar i Harvard-arkitekturen, en för programrutiner och en för datat som ska processas, medan man i von Neumann-arkitekturen skickar allt över samma buss.
(Man har även separataminnen för programrutiner och data i Harvard.)

von Neumann:



Harvard: