

Sammanfattning

Föreläsning 3 - Digitalteknik

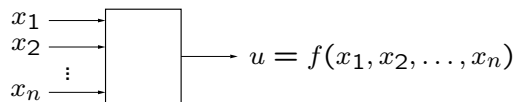
I boken: avsnitt 2.3 (2, 4.1-4.2 i Hemert, Digitala kretsar)

Sekvensnät

En tillståndsmaskin och dess graf är en modell för ett önskat beteende hos ett system. Som en teoretisk modell är den alltså inte bara applicerbar på elektroniska apparater utan kan också modellera mekaniska system, datorprogram och andra informationsflöden. Vi ska nu gå vidare och *realisera* (bygga) vår framtagna maskin med hjälp av enklast möjliga logiska enheter. Dessa nätverk av logiska enheter som realiserar en maskin kallas för *sekvensnät*.

De enklast möjliga logiska enheter som vi kommer att använda i våra sekvensnät är av två typer:

Switchfunktioner eller *Booleska funktioner*. En Boolesk funktion är en funktion med n binära insignaler x_1, x_2, \dots, x_n och en binär utsignal u .



I vår tillämpning kommer in- och utsignal att vara funktioner av den diskreta tiden, $x_1(t), x_2(t), \dots$, men det skrivs oftast inte ut.

De enklaste byggstenarna som ska användas för att göra realiseringen kallas för *grindar*. De fyra viktigaste typerna av grindar är: NOT, AND, OR och Modulo 2 adderare.

NOT grinden, eller inverteraren, ger *komplementet* av insignalen x . Skrivs

$f_{\text{NOT}}(x) = x'$

x	u
0	1
1	0

 $x \rightarrow u = x'$

AND grinden ger utsignal 1 om alla insignaler är 1.

$f_{\text{AND}}(x_1, x_2) = x_1 \wedge x_2 = x_1 x_2$

$x_1 x_2$	$x_1 \wedge x_2$
0 0	0
0 1	0
1 0	0
1 1	1

 $x_1, x_2 \rightarrow u = x_1 \wedge x_2$

OR grinden ger 1 när minst en av insignalerna är 1.

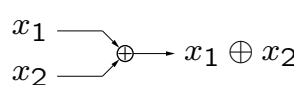
$f_{\text{OR}}(x_1, x_2) = x_1 \vee x_2$
--

$x_1 x_2$	$x_1 \vee x_2$
0 0	0
0 1	1
1 0	1
1 1	1

 $x_1, x_2 \rightarrow u = x_1 \vee x_2$

XOR, Modulo 2 adderare ger utsignal 1 när summan av insignalerna är udda.

$f_{M2A}(x_1, x_2) = x_1 \oplus x_2$	$x_1 x_2$	$x_1 \oplus x_2$
	0 0	0
	0 1	1
	1 0	1
	1 1	0

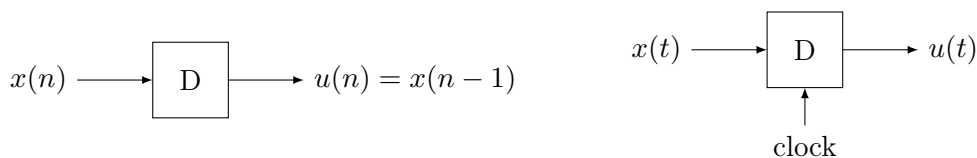


Ett nät med en eller flera Booleska funktioner som inte innehåller återkopplingar (eller fördröjningar) kallas för ett *kombinatoriskt nät*.

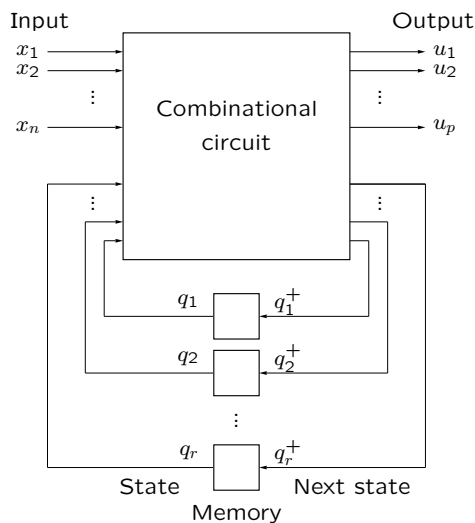
Givet en Boolesk funktion, hur kan jag bygga upp ett nät av grindar som korrekt realiserar funktionen? Det förmodligen enklaste sättet är att konstruera ett 2-nivåers AND-OR nät. Vi använder först n -ingångars AND grindar, som då blir 1 endast för en given insignalkombination. Sedan kombinerar vi dessa med en stor OR grind. Se exempel med majoritetsfunktionen i slides.

Realiseringen av sekvensnät behöver förutom kombinatoriska nät även en annan typ av modul, som kan lagra information. Den enklaste formen av minneselement är det vi kallas för *D-element* (D för delay, fördröjning). Ett D-element har en binär insignal och en utsignal. Utsignalen ges av insignalen fördröjd en tidsenhet, $u(n) = x(n - 1)$.

En *D-vippa* uppdaterar (till värdet av insignalen) beroende på en klocksignal. Det kan exempelvis vara då klocksignalen går från 0 till 1 (positivt flanktriggad).



Givet dessa komponenter är vi nu redo att realisera godtycklig tillståndsmaskin enligt bilden.



Momenten som återstår är att koda tillstånden till en binär representation, ta fram nästa-tillståndsfunktionen och därefter realisera det kombinatoriska nätet med grindar. Se exempel på föreläsningen.