Signals

1) • Synchronous — • Digital

↳ Discrete Values • Discrete Values

2) • Asynchronous — Analog

↳ Continuous. ↳ Continuous.

c) Conversion

Analog ⇒ Digital (A/D)

Digital ⇒ Analog (D/A)

(3) Digital theorem

$f(t) < B$ [Hz] ⇒ Coordinates determined by points spaced $1/2B$ [s] apart.

• State transision graph

• ↳ Visualise state machine.

) • Trellis (LM)

↳ Directed graph representing.

• Parity checking

↳ # 1:or % 2 : i.

(0) Boolean function

is mapping ~~[illegible]~~ to true (1) or false (0)

(1) Sequencial Circuit

is realization of statemachine..

(2) Clock

is updatea verde pe outputum, entut di outputu ir

(3) Euclidis division

$$
\begin{array}{r|l}
 & 0626 \\
\hline
21 & 13151 \\
 & -126 \\
\hline
 & 55 \\
 & -42 \\
\hline
 & 131 \\
 & -126 \\
\hline
 & \boxed{5}
\end{array}
$$

(2) Relativly Primes

$GCD(n_1, n_2) = 1$.

(1) LCM (MGM)

$gcd(n_1, n_2)\, lcm(n_1, n_2) = n_1 n_2$

(2) Euclids recursion

$\gcd(1311, 391)$

$\Leftrightarrow 391 \mid 1311 \Rightarrow$ rest 138

$= \gcd(138, 391)$

$\Leftrightarrow 138 \mid 391 \Rightarrow$ rest 115

$= \gcd(115, 138)$

$\Leftrightarrow 115 \mid 138 \Rightarrow \boxed{\text{rest } 23} \Rightarrow$ GCD

$= \gcd(23, 115)$

$23 \mid 115 = 0$

(3) Bezouts Identity (utilizes Euclids recursion)

$\gcd(n_1, n_2) = s \cdot n_1 + t \cdot n_2$

$23 = 138 - 1 \cdot 115$

$= 138 - (391 - 2 \cdot 138)$

$= \ldots$

(0) Euclids Extended Algorithm

–

(1) Algebraic Structures

↳ Set & one or more operations.

Modulo

↳ $\oplus$, $\otimes$

Groups

↳ en masse krav. (se bok)

Ring

↳ –"–

Commutative Ring

↳ ring and $a \cdot b = b \cdot a$

(3) Unit (invertible element)

$(aa^{-1})_{\text{mod } z} = 1$

Note: $GCD(a, c) = 1 \Rightarrow a$ is invertible.

$\mathbb{Z}_3 \Rightarrow c = 3$.

↳ inverted bezouts algorithm. (why not primes?)

Fields

↳ Commutative ring and $aa^{-1} = 1 \quad \forall\, a \neq 0$

Corollary

↳ $(\mathbb{Z}_p, \oplus, \otimes) \Rightarrow$ field iff $p \in \mathbb{P}$

Indempotent. element.

$a^2 = a$

Boolen ring

↳ $a \in$ Boolen ring && a indempotent.

↳ Theorem.

$(B, \oplus, \otimes)$ boolen ring & $a \in B$

↳ $a + a = 0$ (also $a = -a$)

$(a \wedge b)$

0	0	0
0	1	0
1	0	0
1	1	1

1	1	1	=	0
1	0	1	=	0
0	1	1	-	
0	0	0	=	

Egenskaper; ...

(3) Boolen algebra ⇔ Boolen operations.

$a \wedge b = a \cdot b$

$a \vee b = a + b + ab$ iff $a, b \in$ Boolen Ring

$a' = 1 + a$

(3) Regler

↳ De Morgans

↳ Dualitetsprincipen. $\left(\begin{matrix} 0 \Leftrightarrow 1 \\ \wedge \Leftrightarrow \vee \end{matrix} \right)$

Most significant & least significant bit

$\underline{1}001 0 1 0 1 1 0 1 1 \underline{0}$

MSB LSB.

(3) Full Adder

x_i & y_i är värden
c_i är Carry

$$\begin{array}{r} c_i \\ x_i \\ y_i \\ \hline c_{i+1}\ S_i \end{array}$$

(3) Overflow

$OV = C_n$ (Binär tal) $OV = C_n \oplus C_{n-1}$ (2-comp)

↳ Negativ + Negativ = Positiv
↳ Positiv + Positiv = Negativ.

(2) 2-Complement

-(se bok, stämmer förklaring

↳ Binär
↳ första håller positiv/negativ.
↳ $-x = x' + 1$

$$\underbrace{\begin{aligned} 111 &= -4+2+1 \\ &= -1 \end{aligned}}$$

(1) Multiplikation (?)

```
    1  1  1
    0  1  1
   ---------
    1  1  1
 1  1  1
0 0 0
```

(3) Boolean functions Notation

$B = \{0,1\}$

$\hookrightarrow B^n$ är ett n-dimisonellt uttrymme med 0, 1 i varje dim.

? $\hookrightarrow B_n$ är ett set av alla funktioner från B^n till $\{0,1\}$

$\hookrightarrow B_n^*$ ——— // ——— $\{0,1,-\}$

(3) Inverse Boolean functions.

$\Rightarrow$ (f: input $\Rightarrow$ output)

0: offset of $f^{-1}(0)$

1: onset of $f^{-1}(1)$

–: dontcare set of $f^{-1}(-)$

(2) Latice Exponent

$$x^{(C)} = \begin{cases} 1 : x \in C \\ 0 : x \notin C \end{cases}$$

Note

$x^{(1)} = x$

$x^{(0)} = x'$ (x invers)

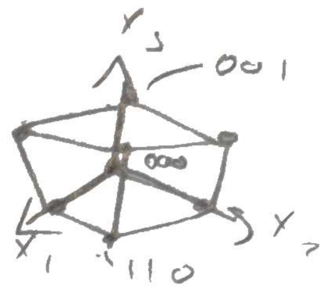
$x^{B} = 1$

$x^{\emptyset} = 0$

(3) Cubes & Cube functions

- $\mathbb{C} = (c_1, c_2, \dots c_n)$, $c_i \in \{\emptyset, 0, 1, B\}$ describes a cube in B^n.
- The cube function is formed by:

$$\mathbb{C}^{\mathbb{C}}(x) = \bigwedge_{i=1}^{n} x_i^{(\mathbb{C}_i)} \quad ; \quad x \in B^n$$



Note:

$c = (B, 0, 1)$

$$\mathbb{C}(x) = C^{(B,0,1)}(x_1, x_2, x_3) = x_1^{(B)} \wedge x_2^{(0)} \wedge x_3^{(1)}$$

↳ Can be translated to

$$\left.\begin{aligned} x_1 &\in B \Rightarrow - \\ x_2 &\in \{0\} \Rightarrow x_2 = 0 \\ x_3 &\in \{1\} \Rightarrow x_3 = 1 \end{aligned}\right\} \Rightarrow x_2' x_3$$

(3) Minterms (describes a 1 corner

↳ a cubefunction of a Vertex (i.e a corner in a cube) equal.

$$m_v = C^{(v)}(x) \qquad \overset{\dots v \dots v \dots}{\underset{\dots \wedge \dots \wedge \dots}{\downarrow}} \qquad x_1 x_2 x_3$$

(3) Maxterms (describes a 0 corner)

The dual of a minterm

$$M_v = (m_v)' = (C^{(v)}(x))' \qquad \overset{\dots \wedge \dots \wedge \dots}{\underset{\dots v \dots v \dots}{\downarrow}} \qquad x_1 \vee x_2 \vee x_3$$

(3) DNF

$f(x) = \bigvee_{a \in f^{-1}(1)} m_a$

(3) CNF

$f(x) = \bigwedge_{a \in f^{-1}(0)} M_a$

(2) Convert to DNF.

Use $a \vee a' = 1$ $\Rightarrow$ $g(x) = x_2 \vee x_1' x_3'$
+ Demorgan

$= (1 \wedge x_2 \wedge 1) \vee (x_1' \vee 1 \vee x_3')$

$= \dots$

(2) Convert to CNF.

Use $0 = a \wedge a'$

Note: CNF contains the "$\wedge$" product of maxterms

$\Rightarrow$ can find DNF $\to$ minterms $\to$ maxterms.

(2) Labelling max & minterms.

I

$\underbrace{(x_1 \vee x_2)}_{11 = 3}(x_1' \vee x_2)$ — $01 = 1$

m_3 m_1

$\boxed{M}$

$(x_1 \vee x_2)(x_1' \vee x_2)$ (Note formatted as maxterms

$((\underbrace{x_1' x_2'}_{01=0}) \vee (\underbrace{x_1 x_2'}_{10=2}))'$

M_0 M_2

Note: M & m can be used to determine if a DNF & CNF are equal.

(3) RMF

↳ Method to write a Boolean function with $\oplus$ & $\otimes$

(3) Derive the RMF

$a \wedge b = ab$

$a \vee b = a \oplus b \oplus ab$ ⇒ generellt

$a' = 1 \oplus a$

Writhe as DNF: then use $m_i \oplus m_j \oplus \underbrace{m_i m_j}$ = $m_i \oplus m_j$

Mintermer, en är alltid $m_i = 0$ eller $m_j = 0$
↳ Dvs de adderar till $\oplus$
↳ kan inte vara samma samtidigt

(2) Function cover

F: Set of cubes with vertices $V(F)$
↳ cubes in F cover f iff

$$f^{-1}(1) \subseteq V(F) \subseteq f^{-1}(1) \cup f^{-1}(-)$$

(3) Implicants

A cube function, $C^c(x)$, for a cube in a cover of f

ie. $C^c(x)$ st $V(c) \subseteq f^{-1}(1) \cup f^{-1}(-)$

(3) Function

if F covers f.

$$f(x) = \bigvee_{c \in F} C^c(x)$$

(3) Minimal Cover

if the number of implicants are minimal, but still covers f

(2) Implicants & covers

- Prime implicant: is not fully covered by another implicant
- Prime cover : consists of only prime implicants
- Essential prime imp: covers a minterm that isn't covered by other implicants.

(3) Find minimal cover (Karnaugh maps)

① Find the E.P.I 's
② find P.I
③ E.P.I + few large P.I

(3) Iterative Consensus

① $ab \vee a'c = ab \vee a'c \vee \underline{bc} = bc$
② $a \vee ab = a$

(1) Equivilence Relation

↳ Reflexive, symmetric, transitive

Equivilence Class

A equivilence relation over S partitions S into disjoint subsets : equivilence classes

(2) State eq.

↳ if ∀ inputs ⇒ output are identical.

(3) RF - algorithm (B) kill?

↳ minimising state machine

① Group states with same output function

② Recursively partition groups based on their next state.

③ Merge states in the same group.

(1) Reachable Graph (5)

A graph if all states are reachable from the starting state

(3) Minimal form

① Remove unreachable states

② Use RF

(2) Finding a good state assignment

(meaning the minimal form is easy to find)

① Reduced dependency (5) dis wer.

↳ Choose a pair of states

↳ Assume the two are in same block

↳ Iteratively ∀ block, group into new group

② 1-Hot

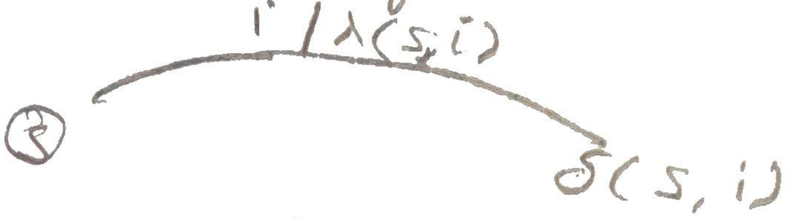
↳ Use one state variable for every state

$S_0 = 1000$, $S_1 = 0100$, $S_2 = 0010$ $S_3 = 0001$

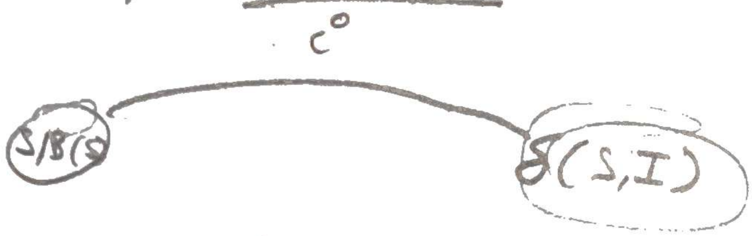
③ Gray Code —

④ Natural Code —

(3) **Mealy** (3)



(3) **Moore** (3)



(2) **Mealy → Moore** (5)

↳ output is delayed one step:

↳ ① Split states s.t. all entering edges have same input.

② Move output into the state. ↳ $S_1 = S_{01}, S_{11}$

(2) **Moore → Mealy** ~~(2)~~ (3)

① Move the output to entering edges

② Use RF

(3) **Stable** (5)

S stable $\iff \delta(s, i) = s$

(3) **Successor states** (4)

If there is a path from S_i to S_{ii}, then S_{ii} is a successor state to S_i for input i.

(3) ~~Asynchronously realizable~~ ~~(7)~~ ?

↳ ~~if all states have successor states~~ ∀ inputs.

(3) Race free State Assignment. (5)

↳ Only one state variable changes when state changes.

↳ Always possible to rewrite asynch to racefree.

(3) Hazard free realization (1) ? sure ?

↳ due to different delays in components, an hazardous output may be given.

Ex $f(x) = x_1 x_3 \vee x_1' x_2'$

x_1 slow, x_1' fast.

$$\left.\begin{array}{l} x_1 = x_2 = 0,\ x_3 = 1 \\ x_1 \to 1 \end{array}\right\} f: 1 \to \underset{\sim}{0} \to 1$$

unwanted hazard.

(3) Solution to hazards

Method 1: Add consensus

Method 2: Use all prime implicants (avoid going over a 0.

Consensus visualized

$x_1 x_2$ \ x_3	0	1
00	1	1
01	0	0
11	0	1
10	0	1

before

added consensus: connects implicants.

(Hazards might occur when we jump out of one prime implicant & into a new.

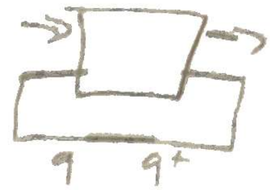
(2) Asynchronous Sequential Circuit.

↳ Not clock controlled

↳ States updates continuously

↳ Mealy-ogi kraum & tap

↳ Moore kan jmp?



(3) Asynchronously realizable (2)

1) ∀ state, ∀ input (∃ succ. state)

2) Hazard free

3) Race free.

~~Var är dessa?~~

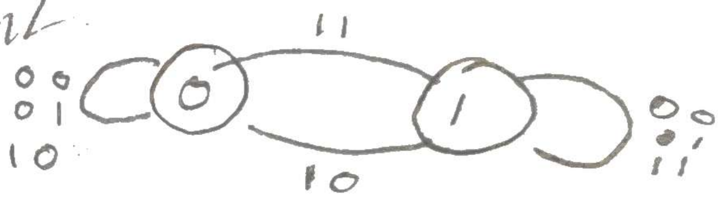
(3) Latch. (5)

↳ Simple memory element

↳ Asynchronous.

input: ϕ x

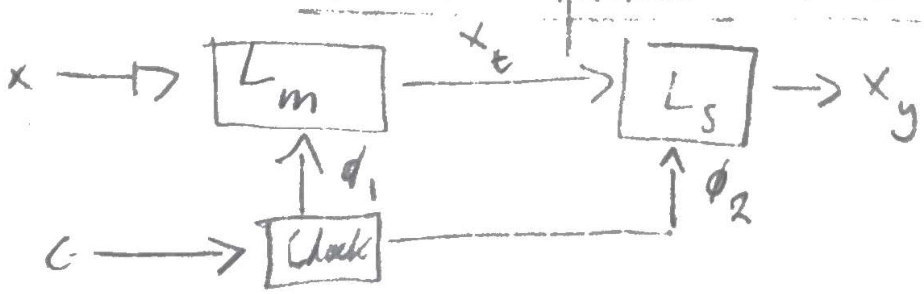
output: $\begin{cases} x & ; \phi = 1 \\ x_0 & \phi = 0 \end{cases}$



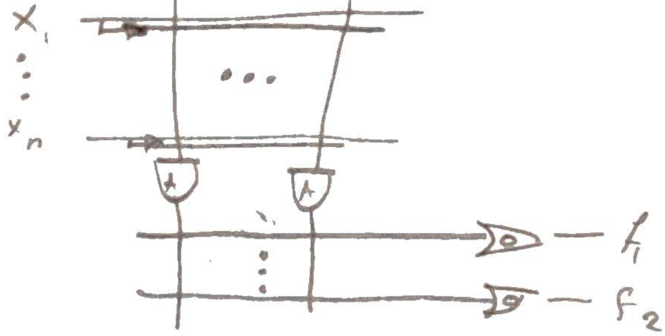
$\phi = 0 \Rightarrow$ stanna / stängd

$\phi = 1 \Rightarrow$ öppen / går till x

D-element implementation (5)



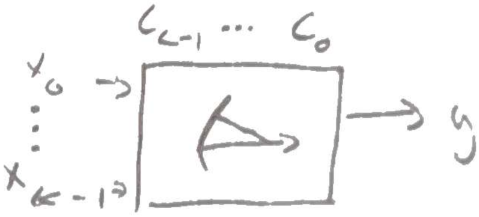
(c) Programable Logic Array



Note: A PLA can also utilize NAND instead of AND & OR

(d) Multiplexer (Switch) (?) Ex? " (5)

↳ A circuit where k bits choose one of 2^k inputs



00001000...

$y = xc = (x_0, x_1, \dots x_{k-1})(c_{k-1}, \dots, c_0)$

Demultiplexer (Decoder) (?) Ex? " (5)

↳ A circuit where k bits choose one of 2^k outputs.

$$y_i = \begin{cases} x & \text{if } \emptyset(c_{k-1} \cdots c_0) = i \\ 0 & \text{else} \end{cases}$$

(e) Tristate

(2) **Linearity** (3) 1

$f(x \oplus y) = f(x) \oplus f(y)$
$f(\alpha x) = \alpha f(x)$

(3) **Linear Boolean function** (0) (1)

A boolean function is linear if it can be written as
$f(x) = a_1 x_1 \oplus \dots \oplus a_n x_n = (a_1, \dots, a_n) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = a\, x$

Visually: Contains n-inputs mod n adders

$\Downarrow$

$q^+ = A q + B x$
$z = C q + H x$

(3) Linearity test

(3) **Rank** (0)

... like extra ...

↳ Divided by gauss elimination

(3) Diagnostic Matrix (5)

$pr \times r$ matrix $(r = \# q_n)$
$(p = \# up)$

$$K = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{pmatrix}$$

Note if $\mathrm{Rank}(K) < \min(pr, r)$
↳ Can be simplified.

(3) Reduced form (simplifying the DM) (5)

$A_{reduced} = TAR$
$B_{red} = TB$
$C_{red} = CR$
$H_{red} = H$

T: first k linearly independent rows of the DM

R: Right inverse of T
↳ $TR = I$

(3) D - Tranform

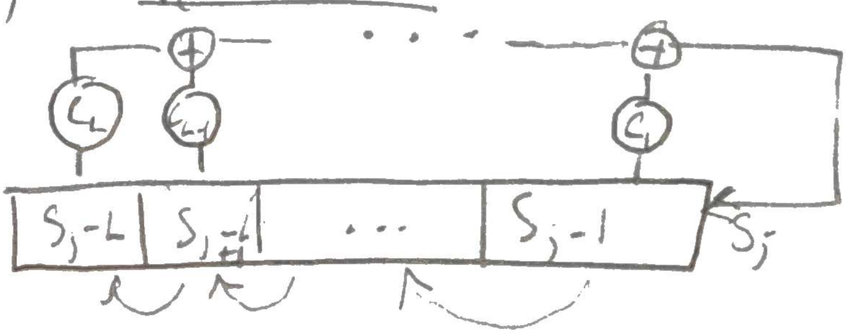
$x = \dots x_{-1} x_0 x_1 \dots$ can be represented by $x(D)$
delayed one time instance.

$x(D) = \dots \oplus x_{-1} D^{-1} \oplus x_0 \oplus x_1 D \oplus \dots$

(3) LFSR Theorem (Linear feedback shift register) (5)

- $C(D)$ connection polynomial
- $S(D) = \dfrac{P(D)}{C(D)}$ $P(D)$ → Determines Starting State. $C(D)$ → Determines feedback.
- Find the shortest LFSR, $S(D)$: simplify $\frac{P(D)}{C(D)}$ using Euclids algo

(3) Realization



Shifts with clock.

(2) Maximal Length Sequence 55)

↳ All LFSR give seperate cycles

↳ if ∀ States ($\neq 0$) are in the cycle

$\Longrightarrow$ output is a maximal length sequence

(3) Regarding $C(D)$:

$$C(D) = 1 \oplus \sum_{i=1}^{L} c_i D^i \overset{c_0=1}{=} \sum_{i=0}^{L} c_i D^i$$

on LSFR circuit

Ex:

$C(D) = 1 + D + D^4$

$\Longrightarrow$ Used to build the LSFR

(3) Starting State ($P(D) \Rightarrow S$)

$$S(D) = \frac{P(D)}{C(D)} = \text{Starting state}$$

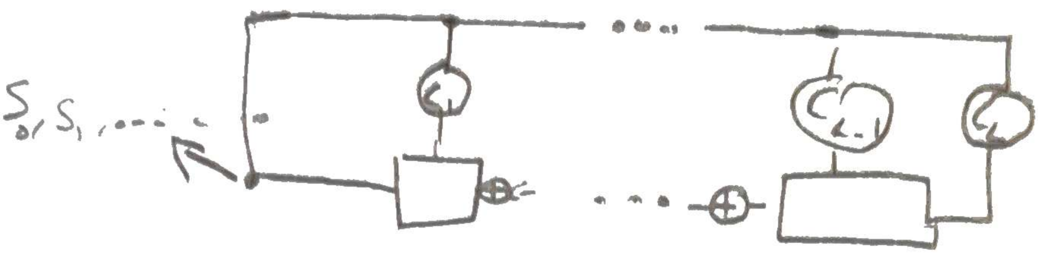
(3) Starting State ($S \Rightarrow P(D)$)

$$S(D) = \frac{P(D)}{C(D)} \Rightarrow P(D) = S(D)\,C(D) \text{ iff } \deg P(D) < \deg C(D)$$

Always achievable.

(3) <u>Observer Canonical form of LFSR</u> (2)

↳ Alternative way of realization



(4) <u>Regarding $S(D) = \frac{P(D)}{C(D)}$</u> (see earlier diagram)

↳ The remainder of each step of the division is the states delayed by the next transform

Meaning: Write down the cycle of states. These will correspond to the remainders.

$111 \Rightarrow 1 \oplus D \oplus D^2$

$010 \Rightarrow D$

↳ ...

<u>The Period of C(D)</u>

↳ Smallest $T > 0$, which $C(D) \mid 1 + D^T$

$\Rightarrow$ Derived by $\frac{1}{C(D)}$ and first remainder $D^k \Rightarrow T = k$

Note $C(D)_{period} = S(D)_{period}$ if $\gcd(P(D), C(D)) = 1$

C(D): ~~antalet states~~ & hur de är ihopkopplade

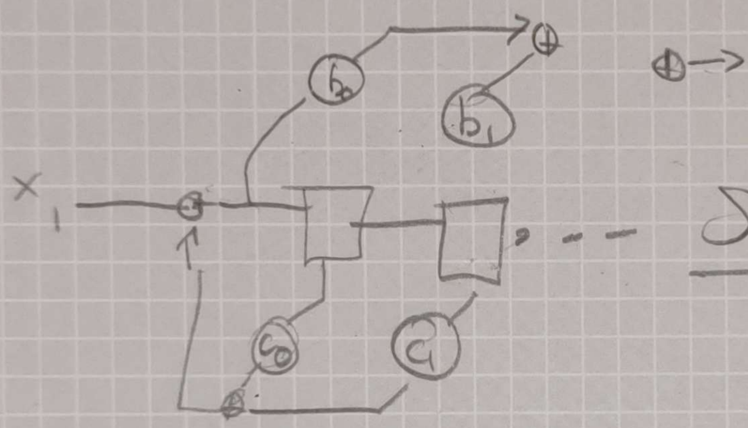
S(D): Series. De n första (se C(D)) är slutliga

P(D): transformerar från C(D) till S(D)

↳ Visuellt? hur?

Observer Canonical form (styrbar)

$$g(D) = \frac{b(D)}{c(D)} = \frac{b_0 \oplus b_1 D \oplus \cdots \oplus b_m D^m}{1 \oplus \cdots \quad c_m D^m}$$



Se bok.

; $g(D)$

Observable

$g(D) = -||$

... se bok

$g(D)$ relaterar mellan x & u.

↳ $u(D) = G(D)x(D)$

$$G(D) = C(I \oplus AD)^{-1} BD \oplus H$$