Class Notes
for EE 3755
Part II

Prepared by: Alexander Skavsntzos

- The number of elements in a vector is called dimension of the vector

For example, a 32-bit register A has dimension 32.

- Naming the elements of a vector

Let A be a 32-bit vector

Then A = A[0], A[1], A[2], ..., A[31]

The above naming scheme is called

0-origin indexing

or

0-origin counting scheme.

- A[5:10] = A[5], A[6], A[7], A[8], A[9], A[10]

- Matrices (2-dimensional arrays of operands)

Below you see the components of a k-row by n-column matrix (or a matrix of dimension k-by-n)

M<0>[0]	M<0>[1]	M<0>[2]	....	M<0>[n-1]
M<1>[0]	M<1>[1]	M<1>[2]	....	M<1>[n-1]
M<2>[0]	M<2>[1]	M<2>[2]	...	M<2>[n-1]
⋮				⋮
M<k-1>[0]	M<k-1>[1]	M<k-1>[2]	...	M<k-1>[n-1]

---

row index ← M<i>[j] → column index

- M<i>[j] = component of row i and column j of matrix M
- M<i:j> = matrix consisting of rows i through j (included) of matrix M.
- M[m:q] = matrix consisting of columns m through q (included) of matrix M.

Boolean Operators

NOT denoted by $\overline{\ }$; AND denoted by $\wedge$; OR denoted by $\vee$; Exclusive OR (EX-OR) denoted by $\oplus$

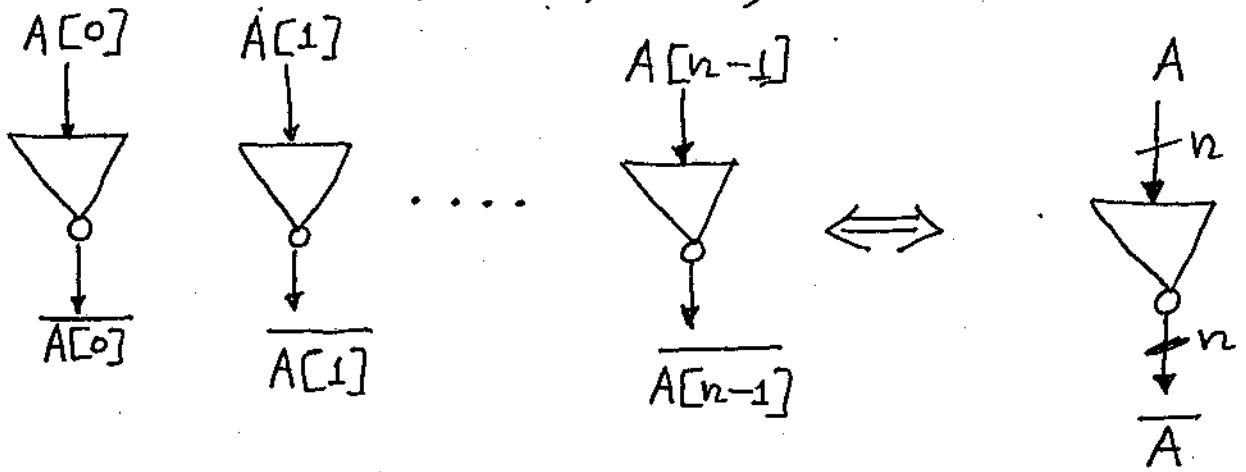
- The operators $\wedge$ $\vee$ $\oplus$ apply between vectors or matrices of the same dimension except in case of vectors (matrices) and a scalar.
- The operator NOT applies componentwise on the elements of a vector or a matrix

Example: Consider two n-bit vectors

$A = A[0], A[1], A[2], \cdots, A[n-1]$

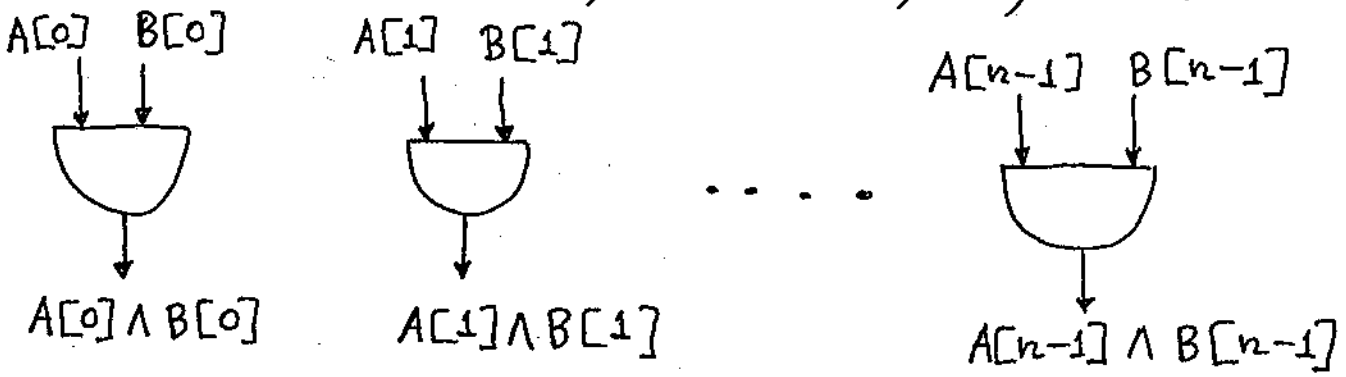
$B = B[0], B[1], B[2], \cdots, B[n-1]$

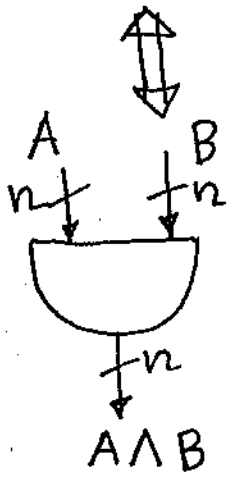
Then $\overline{A} = \overline{A[0]}, \overline{A[1]}, \cdots, \overline{A[n-1]}$



(4) a

$$A \wedge B = (A[0] \wedge B[0],\ A[1] \wedge B[1],\ \ldots,\ A[n-1] \wedge B[n-1])$$





$$A \vee B = (A[0] \vee B[0],\ A[1] \vee B[1],\ \ldots,\ A[n-1] \vee B[n-1])$$

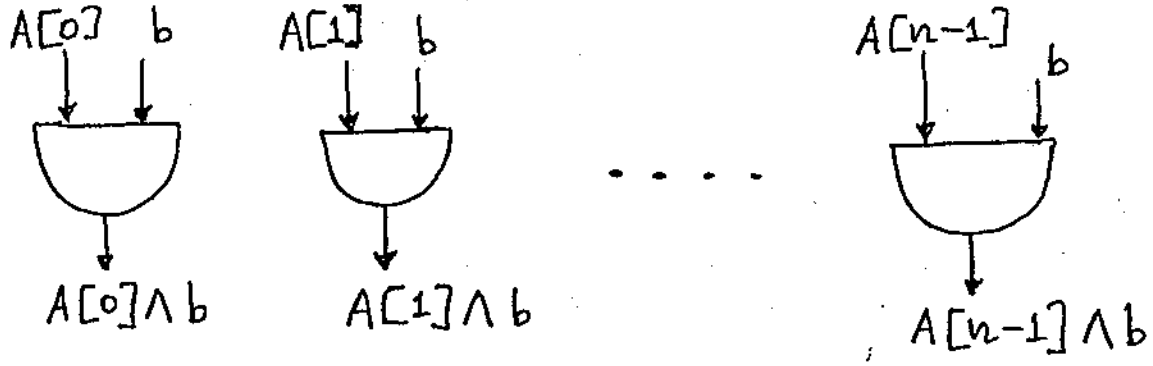
- The situation is similar with the EX-OR

(5)a

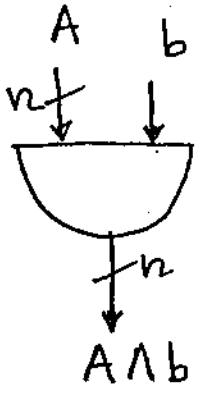
Let b be a scalar and A be a vector of dimension n or $A = A[0], A[1], \ldots, A[n-1]$.

Then

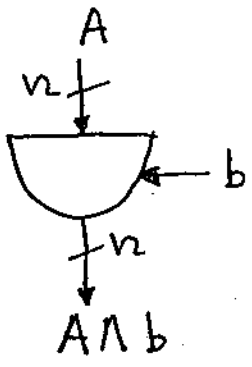
$$A \wedge b = (A[0] \wedge b, A[1] \wedge b, A[2] \wedge b, \ldots, A[n-1] \wedge b)$$



or



or



The situation is similar with $A \vee b$ and $A \oplus b$.

$A[10:12] \wedge B[12:14] =$

$= A[10] \wedge B[12], A[11] \wedge B[13], A[12] \wedge B[14]$.

• Catenate operation

The catenate operation is denoted by , (comma). Let X and Y be the vectors

$X = X[0], X[1], \cdots, X[n_1 - 1]$

$Y = Y[0], Y[1], \cdots, Y[n_2 - 1]$.

Then

$X, Y = X[0], X[1], \cdots, X[n_1 - 1], Y[0], Y[1], \cdots, Y[n_2 - 1]$.

Example: Consider a 4-bit vector A and an 8-bit vector B. Then

$A[2:3], B[2:4], B[6:7] =$

$= A[2], A[3], B[2], B[3], B[4], B[6], B[7]$.

• Row catenate (it applies on matrices)

The notation for row catenation is !

Let M and N be matrices. Then

M ! N = a new matrix with the rows of M above the rows of N.

Example: $M = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$; $N = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$.

Then

$$M\,!\,N = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

- Reduction for vectors

Let X be a vector of dimension n

$X = X[0], X[1], X[2], \ldots, X[n-1]$.

Reduction by means of operation $\odot$ applied on vector $X =$

$= \odot/X = X[0] \odot X[1] \odot X[2] \odot \cdots \odot X[n-1]$

where $\odot$ is $\vee$ $\wedge$ $\oplus$

Example: $X = X[0], X[1], X[2], X[3]$.

Then

$\wedge/X = X[0] \wedge X[1] \wedge X[2] \wedge X[3]$.

The situation is similar with the operators $\vee$ $\oplus$.

• Reduction applied on matrices

Let M be a matrix

- Row reduction by means of $\odot$ applied on M $= \odot/M =$ horizontal vector where each component of the vector is the result of reducing a row by means of operation $\odot$
- Column reduction by means of $\odot$ applied on M $= \odot//M =$ horizontal vector where each component is the result of reducing a column by means of the operation $\odot$

Again $\odot$ can be $\vee$ $\wedge$ $\oplus$

Example: Let N be $N = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Then $\wedge/N = (0, 1, 0, 1)$

$\oplus//N = (1, 1, 0)$

$\vee/(\oplus//N) = \vee/(1,1,0) = 1 \vee 1 \vee 0 = 1$

- Binary encode

The notation for binary encoding is T

nTp = n-bit vector representation of value p.

For example

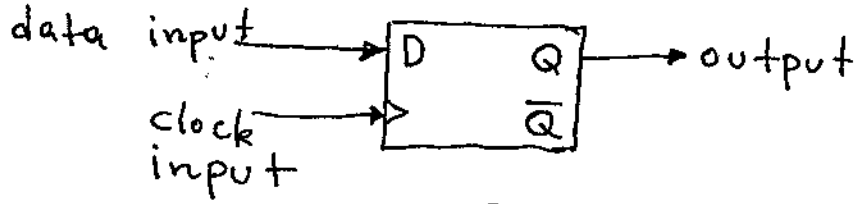
3T0 = 0,0,0

4T10 = 1,0,1,0

5T15 = 0,1,1,1,1

7T32 = 0,1,0,0,0,0,0

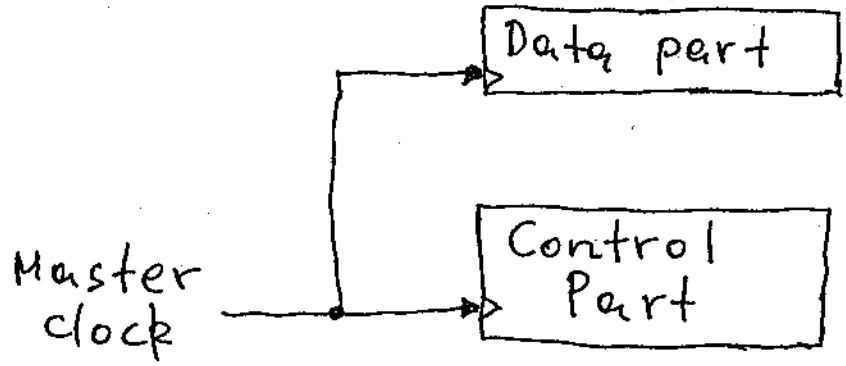
$\overline{4T0}$ = $\overline{0,0,0,0}$ = 1,1,1,1

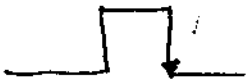


D Flip Flop (FF)

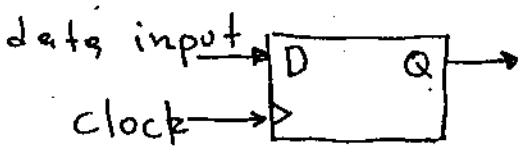
Register transfers are clocked which means that they are synchronized by a system master clock.

We assume that the control section as well as the data section of a digital system are synchronized by the system master clock



Changes of states of flip flops (FFs) can happen at the leading edge of a clock pulse  or at the trailing edge of a clock pulse 

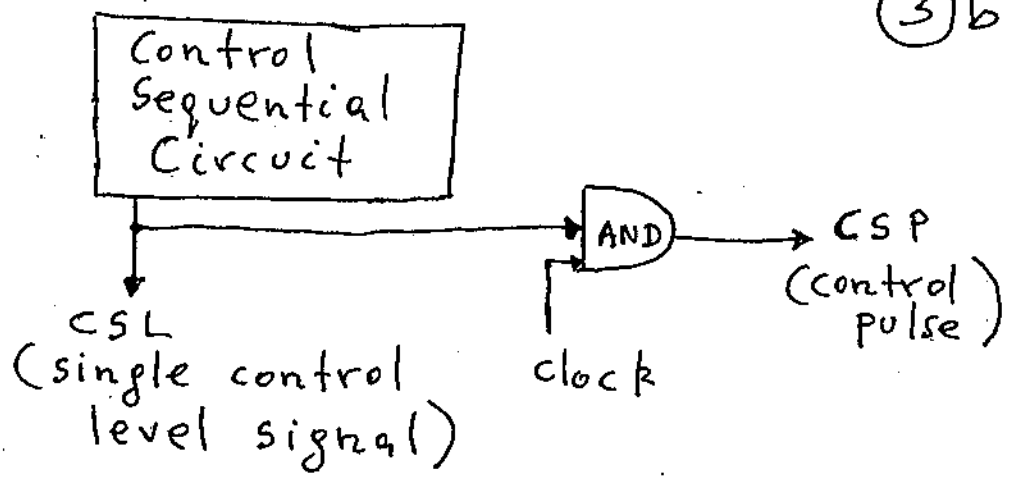
We will assume trailing-edge-triggered systems; (this means trailing-edge triggered flip-flops will be assumed).

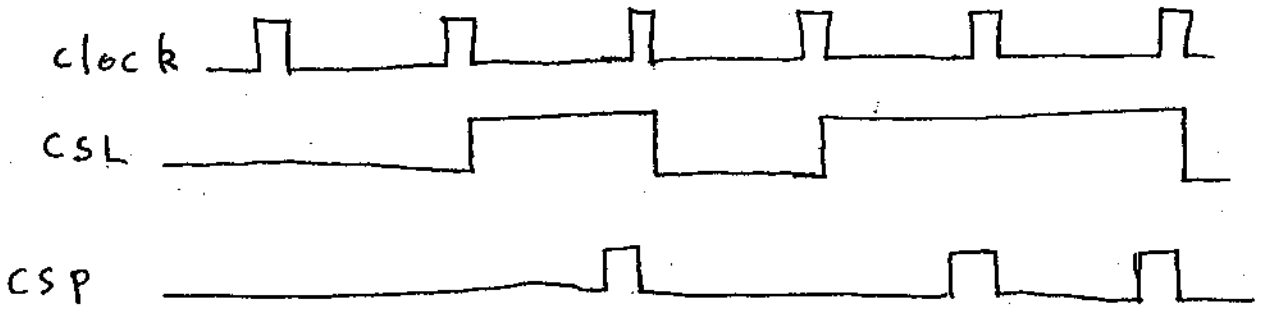


Whatever is the D input it will be stored in Q right after 

↗ trailing edge of the clock.

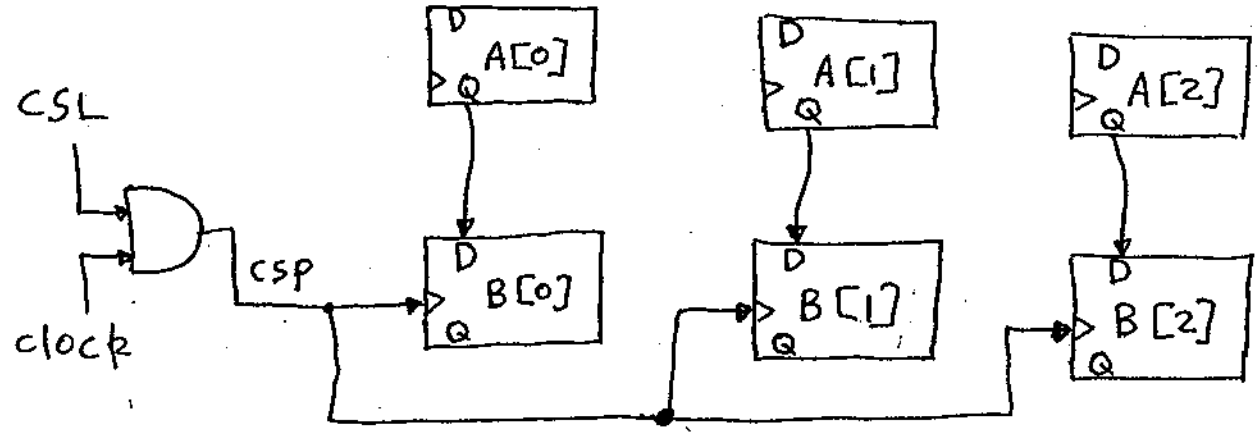
(3) b


Control
Sequential
Circuit
AND
CSP
(control
pulse)
CSL
(single control
level signal)
clock

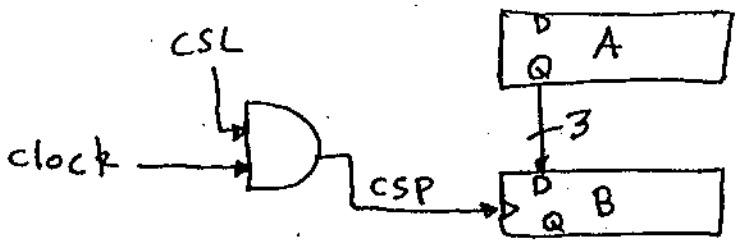

clock
CSL
CSP

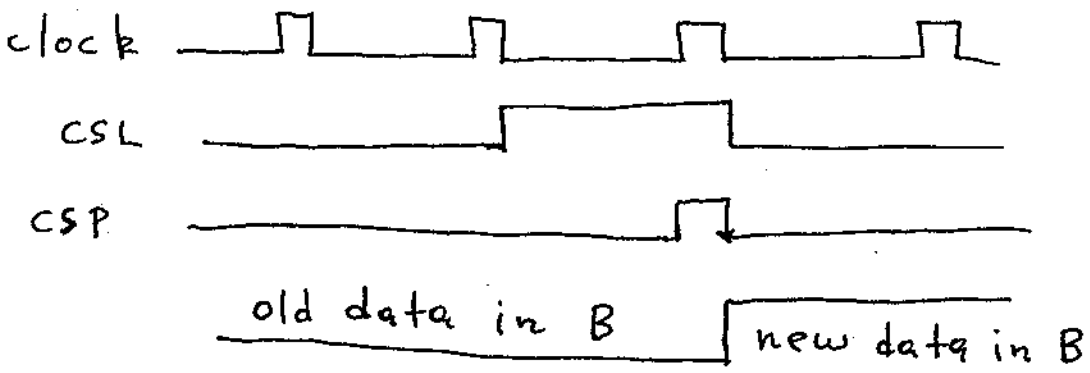
(4) b

- A and B are 3-bit registers. Do B ← A if CSL = 1. Show in logic block diagram form the above transaction.



or





(5) b

- A B C D are 3-bit registers.

Do

D ← A if CSL1 = 1

D ← B if CSL2 = 1

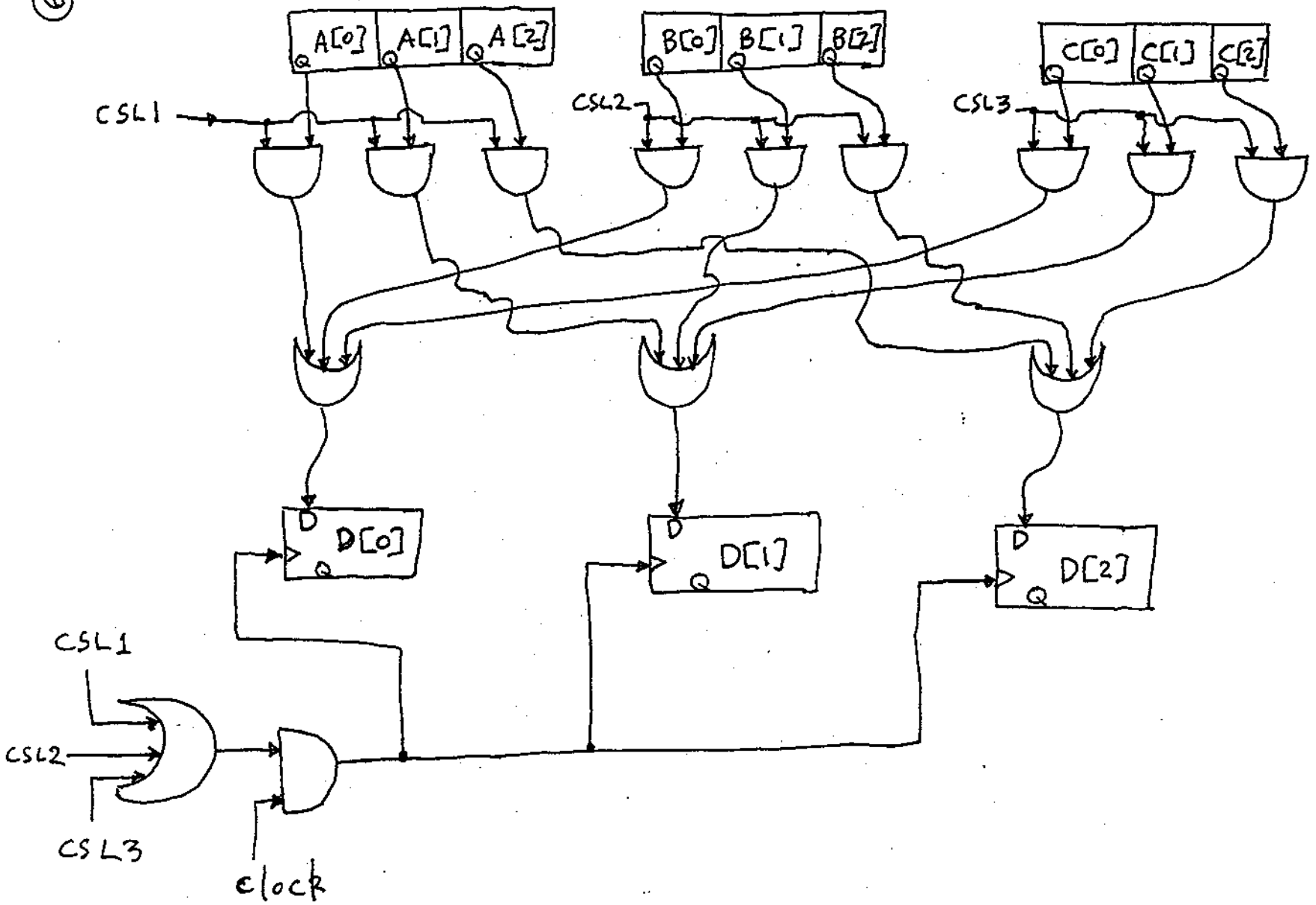
D ← C if CSL3 = 1.

Question: Is there any restriction regarding the control signals CSL1, CSL2, CSL3 ?

Answer: At most one of CSL1 CSL2 CSL3 can be 1.

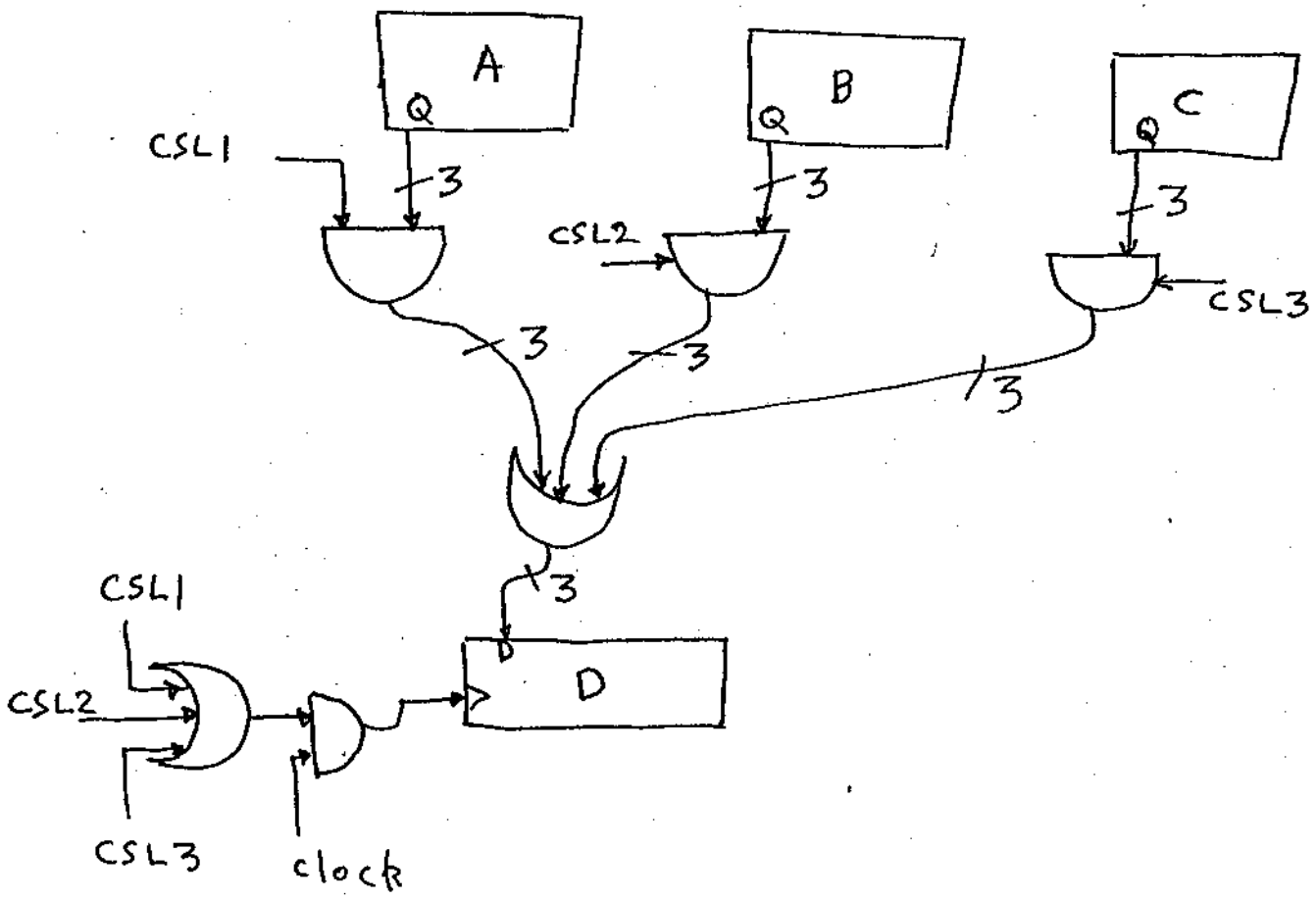
- Show in block diagram form the above transactions.

9 ⑨


A[0]
A[1]
A[2]
B[0]
B[1]
B[2]
C[0]
C[1]
C[2]
CSL1
CSL2
CSL3
D
D[0]
D[1]
D[2]
Q
CSL1
CSL2
CSL3
clock

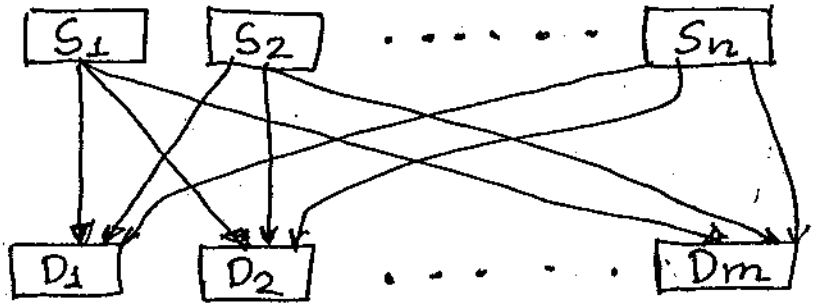
(7) b

The bit level diagram of the previous page is now shown in a more compact way



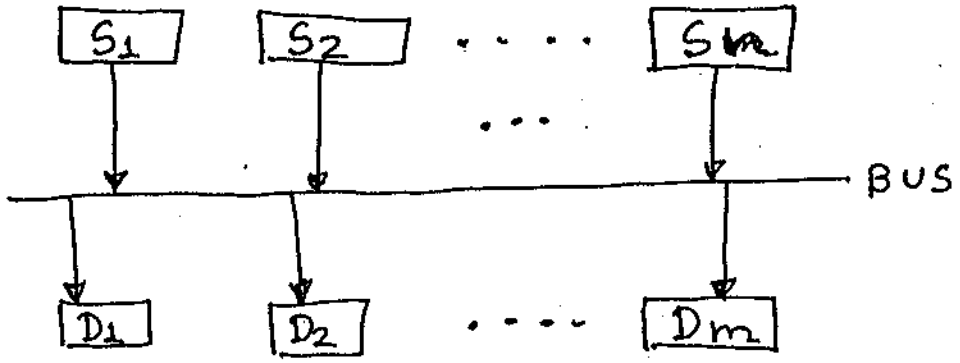
- The general problem:
- Connect n sources (origin vectors) to m destinations.
- Two approaches can be followed:

(a) Using dedicated data paths



n x m dedicated data paths

(b) Using a single BUS



Only one source S_i can be connected to the BUS at any time

** We will compare the cost/speed of the two above approaches.

Example of approach @; (dedicated data paths).

A; B; C are b-bit source (origin) registers.
D; E are b-bit destination registers.

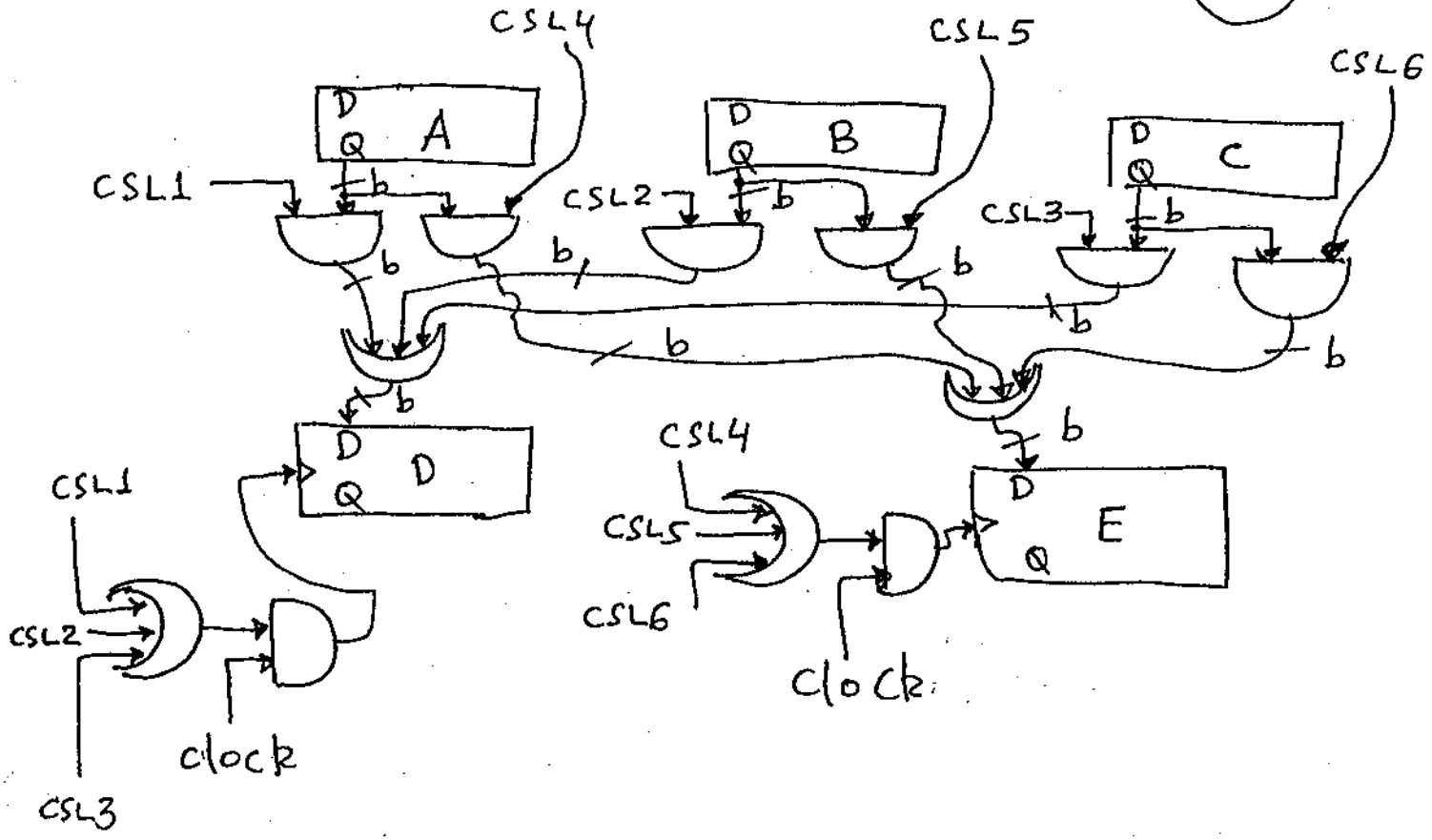
Do
D ← A if CSL1 = 1
D ← B if CSL2 = 1
D ← C if CSL3 = 1
E ← A if CSL4 = 1
E ← B if CSL5 = 1
E ← C if CSL6 = 1.

Question: Is there any restriction regarding the control signals CSL1, CSL2, ..., CSL6 ?

Answer: At most one of CSL1, CSL2, CSL3 can be 1. Also, at most one of CSL4, CSL5, CSL6 can be 1.

- Show in logic block diagram form the above transactions.

(10) b



Here the cost for connecting A; B; C to D; E is

6 groups (or banks) of AND gates
+) 2 groups (or banks) of OR gates

8 groups of AND/OR gates
or 8×b bit-level AND/OR gates.

In general: If we have n sources (origin vectors) and m destinations, the cost of connecting any source to any destination using the dedicated data path approach is

$n \times m$ groups (banks) of AND gates
+) m groups (banks) of OR gates

$n \times m + m = m(n+1)$ groups of AND/OR gates

or

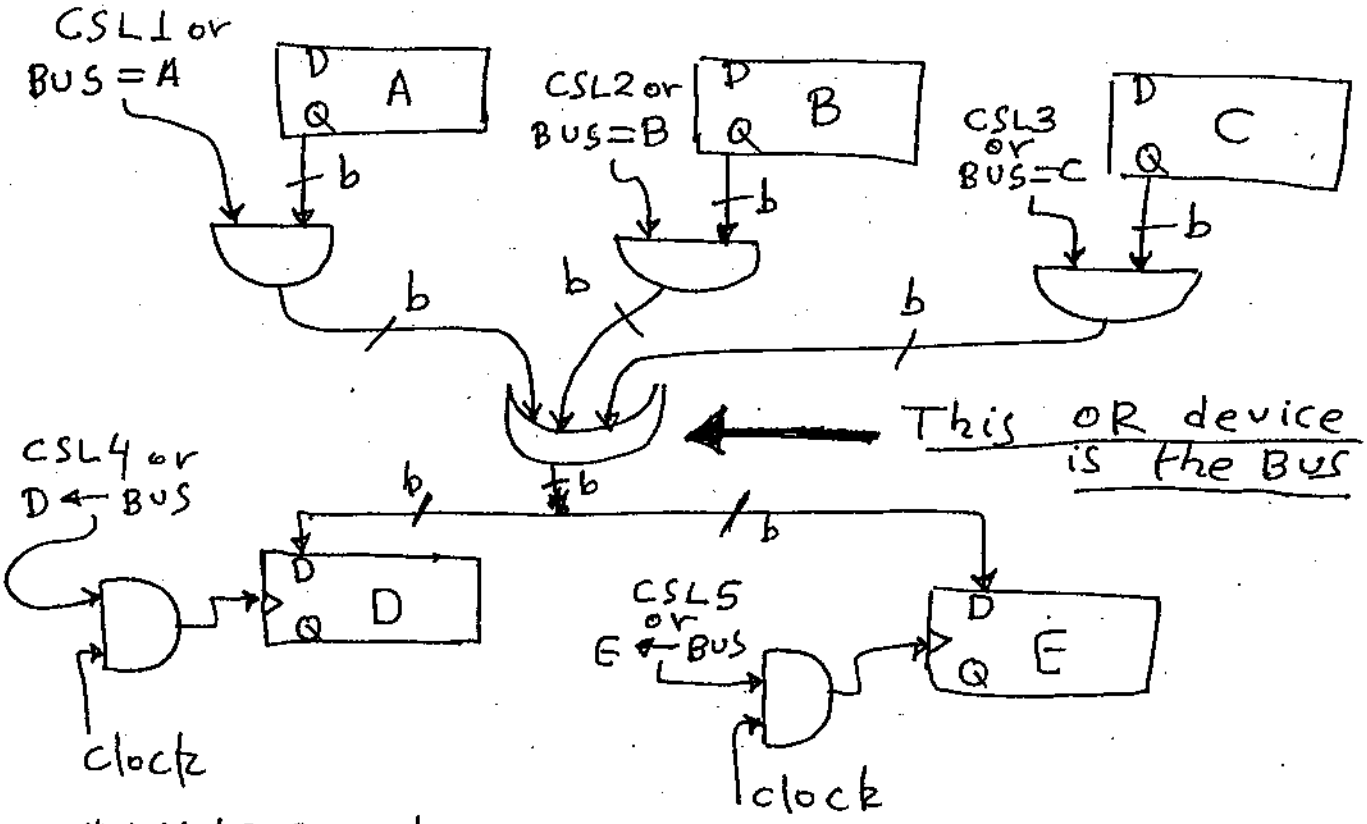
$$\text{Cost}_{\text{dedicated}} = m(n+1) \cdot b \quad \begin{matrix} \text{bit level} \\ \text{AND/OR gates} \end{matrix} \quad (1)$$

where b = number of bits per source or destination.

Example of approach (b); (single BUS).

A; B; C are b-bit source (origin) registers. D; E are b-bit destination registers.

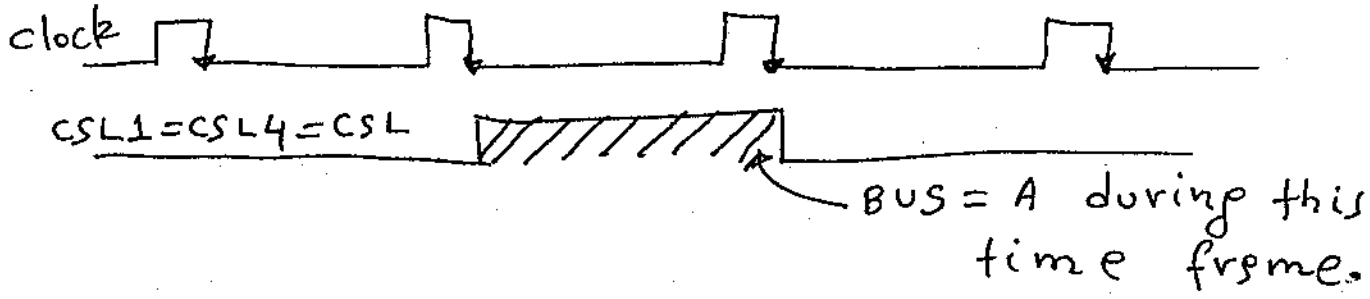
- Show in logic block diagram form how any transfer from any source to any destination can be mechanized using a single BUS.

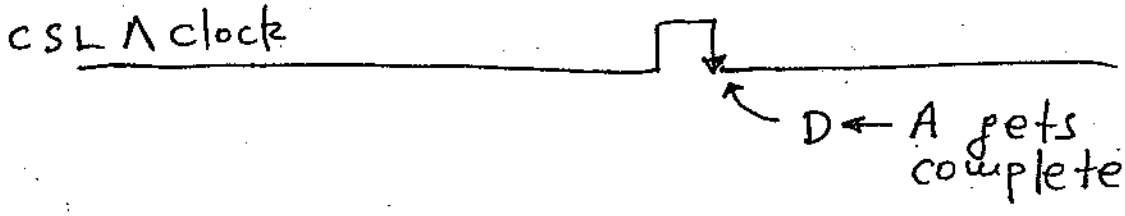


** Note: Only one of CSL1, CSL2, CSL3 can be true at any time.

Timing diagram:

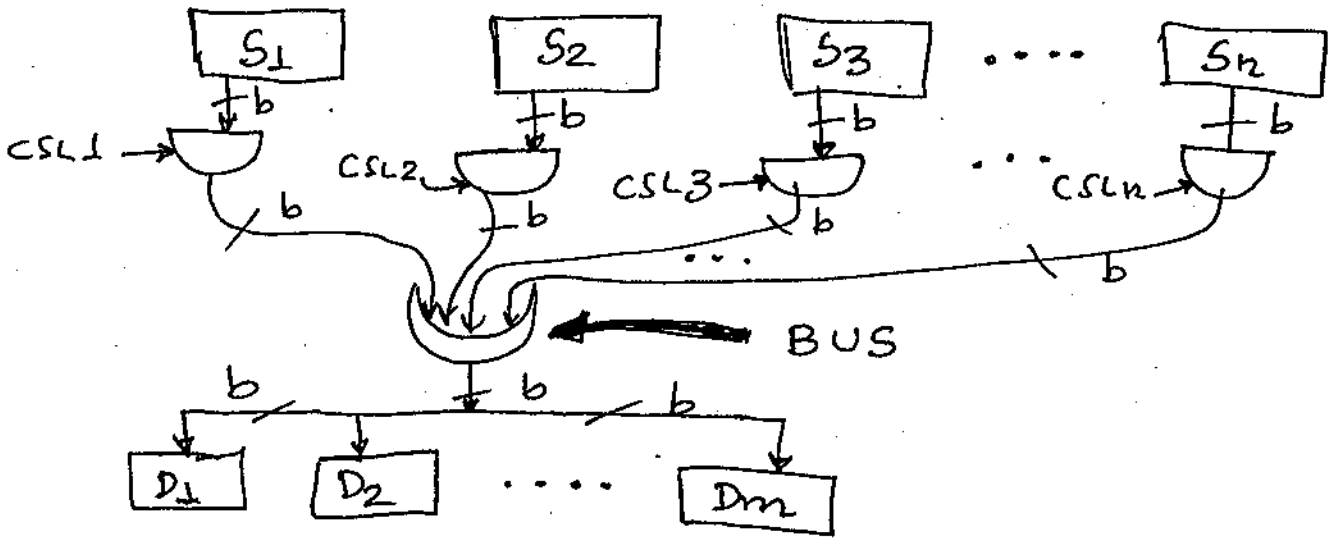
Suppose D ← A needs to be accomplished.





In general:

n sources (origin vectors);
m destinations;
b bits per source/destination



(14).b

The cost of connecting any source to any destination using the BUS approach is

n groups of AND gates
+) 1 group of OR gates; (the BUS)

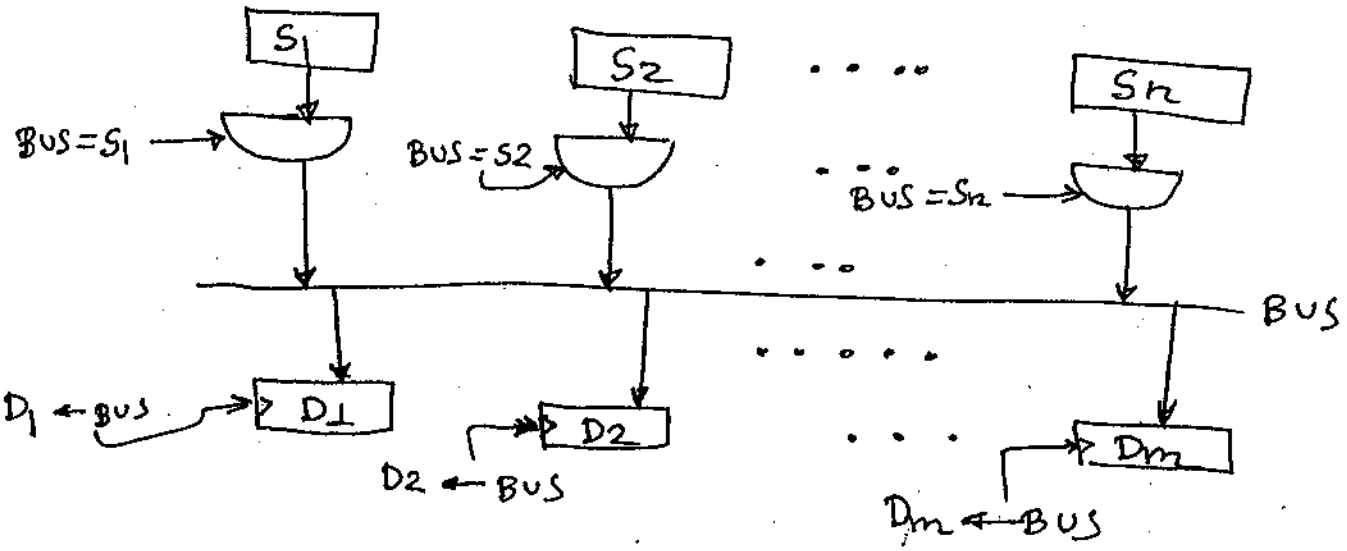
n+1 groups of AND/OR gates

or

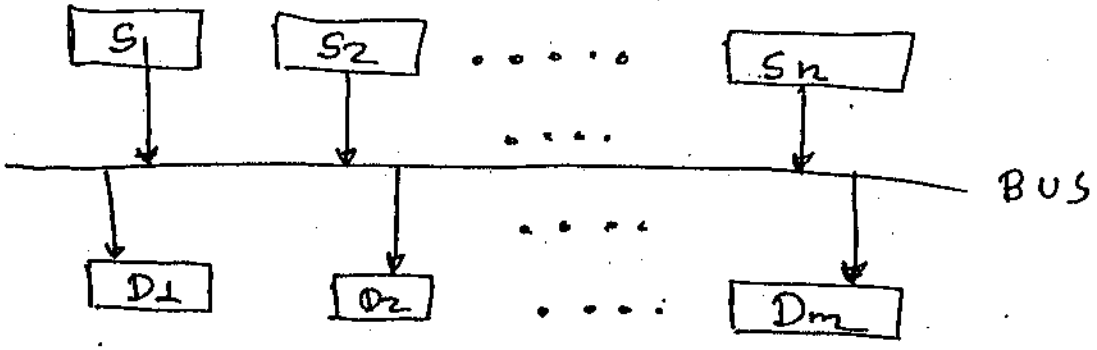
$$Cost_{BUS\text{-}connect.} = (n+1) \cdot b \text{ bit level AND/OR gates} \quad (2).$$

* Compare costs of (1), (2).
  Tradeoff: cost vs. speed.

Notations for BUS connections

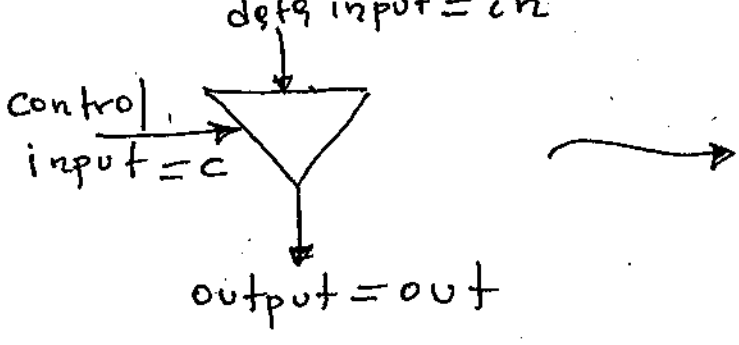


or



- Another approach for constructing long BUSES (buses used to connect digital systems being far away from each other) is by using tristate elements.

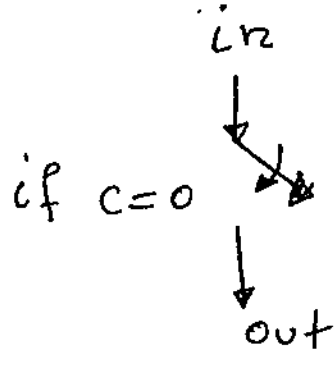
- Tristate element



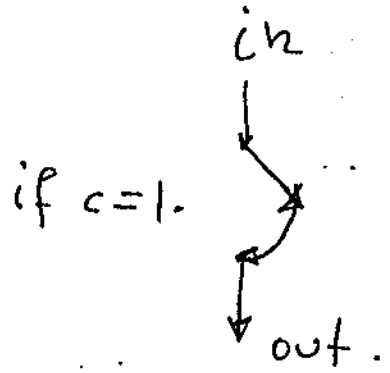
if c=0 ⇒ out/in isolated; (open switch).

if c=1 ⇒ out = in; (closed switch).

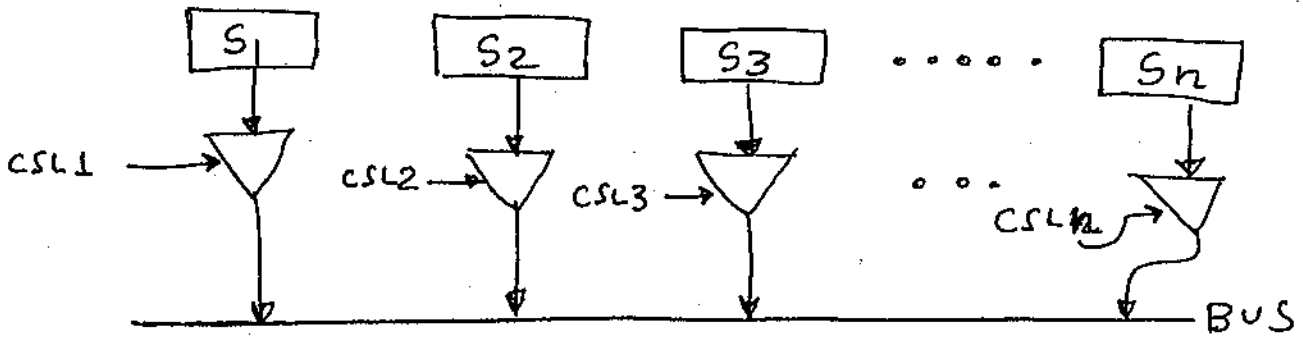
OR



(high impedence).

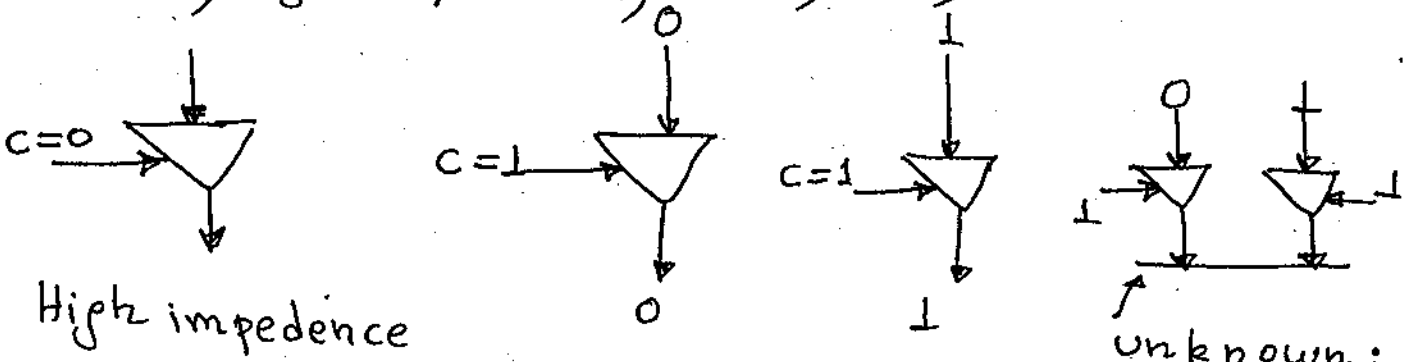


• Tristate BUSES



*: At most one of CSL1, CSL2, ..., CSLn can be 1.

**: Tristate elements can get in four different states; high impedence; zero; one; unknown.



High impedence

unknown; (must be avoided).