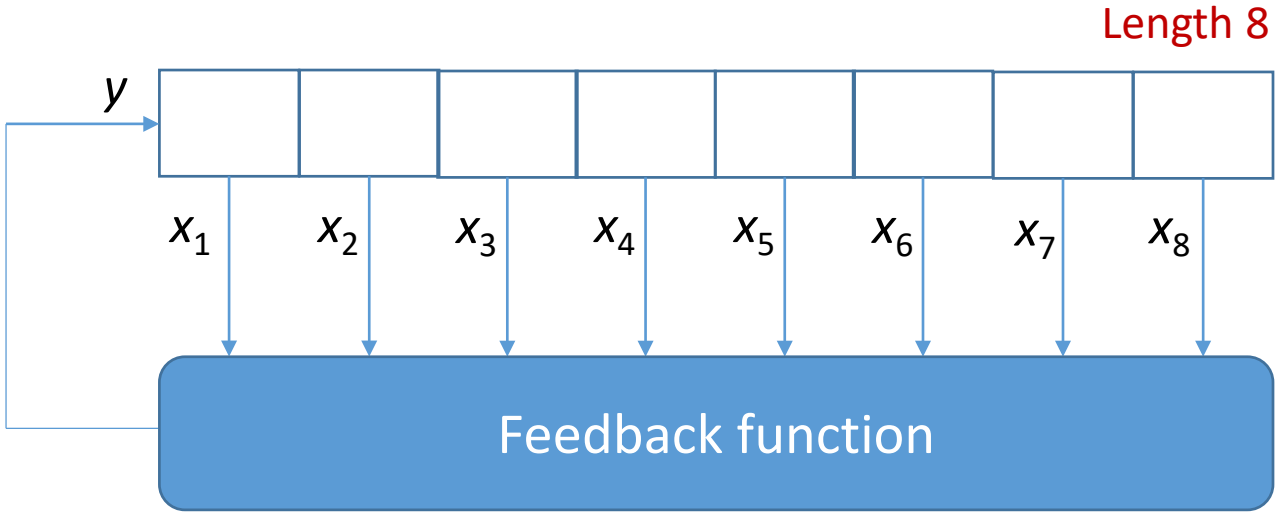
Digital Electronic Circuits
Section 1 (EE, IE)

Lecture 24

Feedback Shift Register



Linear feedback: $y = C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4 + C_5x_5 + C_6x_6 + C_7x_7 + C_8x_8$

C_i = 0 or 1 ⇨ when 1, the output bit is tapped

+ : Sum operation obtained by Ex-OR

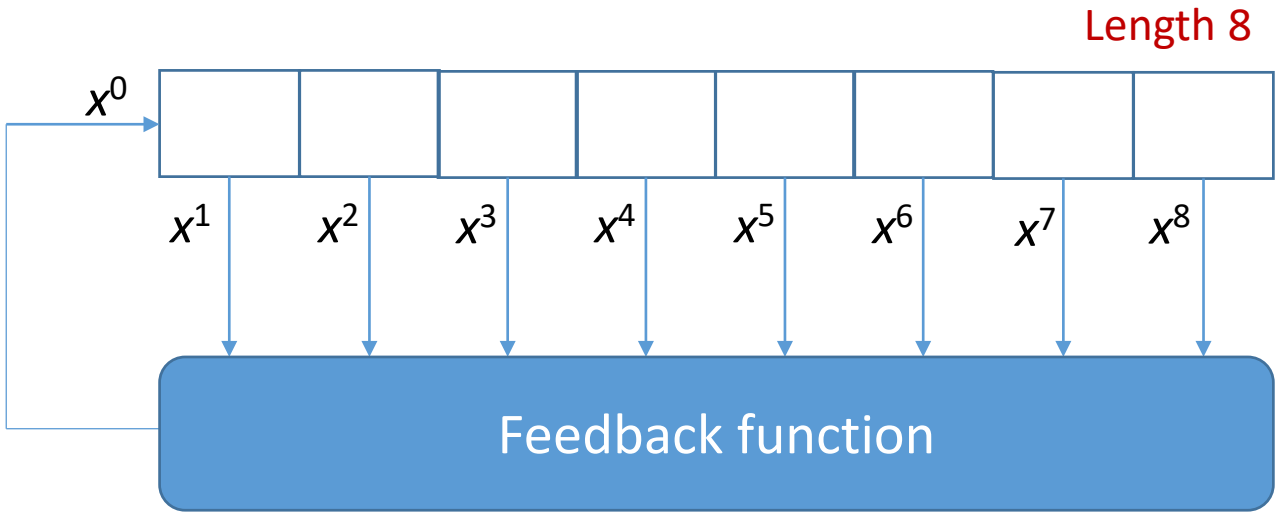
Linear feedback example:

$y = x_1 + x_2 + x_7$

If shift register (SR) contains,
10011011 → $y = 1 + 0 + 1 = 0$
With clock trigger, SR value
0100110**1** → $y = 0 + 1 + 0 = 1$
…

Nonlinear feedback example: $y = x_1x_2 + x_7$

Feedback Polynomial



Example:

$f(x) = x^8 + x^7 + x^4 + x^2 + x + 1$

$f(x) = x^8 + x^7 + x^2 + x + 1$

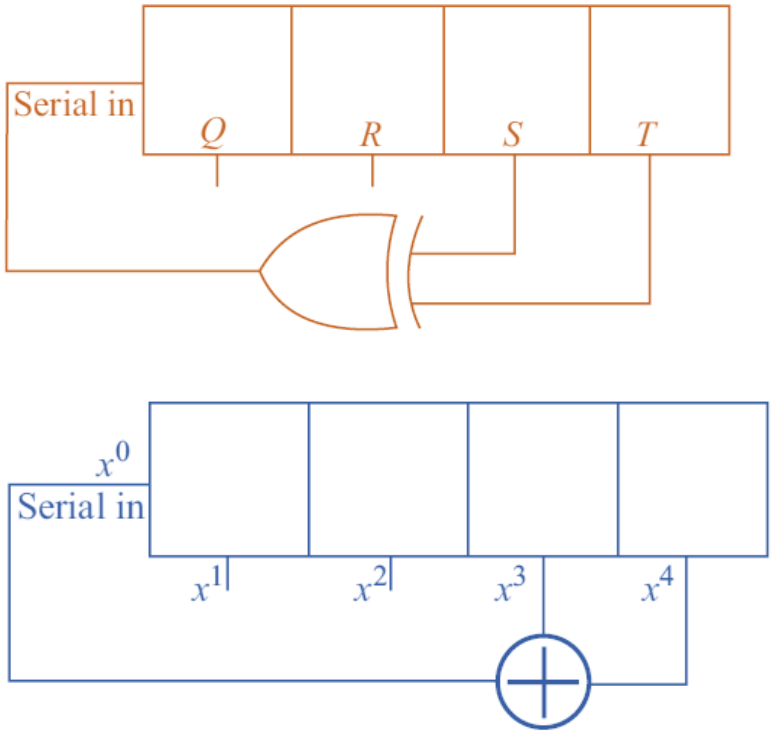
$\boldsymbol{f(x) = x^8 + x^7 + 1}$

Tap from bit 7 and 8 Ex-Ored and fed as serial input to bit 1

$$f(x) = 1 + C_1x^1 + C_2x^2 + C_3x^3 + C_4x^4 + C_5x^5 + C_6x^6 + C_7x^7 + C_8x^8$$

For *n*-bit shift register, degree of the polynomial is *n.*

Pseudorandom Sequence



Q	R	S	T	Serial in $= S \oplus T$	Clock cycle
1	1	1	1	0	1
0	1	1	1	0	2
0	0	1	1	0	3
0	0	0	1	1	4
1	0	0	0	0	5
0	1	0	0	0	6
0	0	1	0	1	7
1	0	0	1	1	8
1	1	0	0	0	9
0	1	1	0	1	10
1	0	1	1	0	11
0	1	0	1	1	12
1	0	1	0	1	13
1	1	0	1	1	14
1	1	1	0	1	15
1	1	1	1	0	16 (repeats)

$f(x) = x^4 + x^3 + 1$

Cycle length = 15

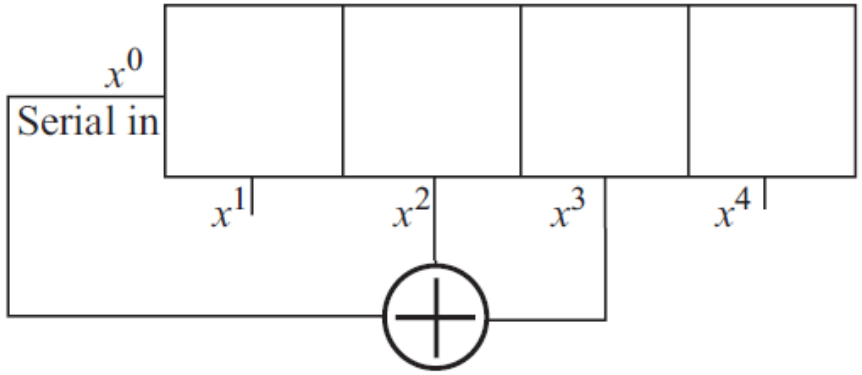
Pseudorandom sequence:
000100110101111..

If *QRST* = 0000, it remains locked i.e. no change in state

Also possible with Ex-NOR feedback where 1111 excluded.

Non-Maximal Length

Q	R	S	T	Serial in $= R \oplus S$	Clock cycle
1	1	1	1	0	1
0	1	1	1	0	2
0	0	1	1	1	3
1	0	0	1	0	4
0	1	0	0	1	5
1	0	1	0	1	6
1	1	0	1	1	7
1	1	1	0	0	8
0	1	1	1	0	9



$f(x) = x^3 + x^2 + 1$

Cycle length = 7

Primitive Polynomials

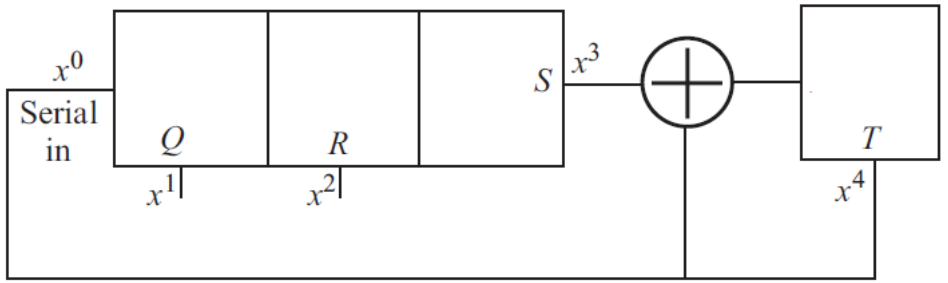
- Polynomials that produce maximal length ($2^n - 1$) sequence are called primitive polynomials.
- Necessary (but, not sufficient condition) to be primitive polynomial
  - No. of taps even
  - Tap numbers are co-prime
- If tap sequence of *n*-bit LFSR generating primitive polynomial is ***n, m, l, k, …, 0*** then the tap sequence *n - n, n - m, n - l, n - k, …, n -* 0 i.e. **0, *n - m, n - l, n - k, …, n*** will also give primitive polynomial.

Degree	Polynomial[#]
2, 3, 4, 6, 7	$x^n + x + 1$
5	$x^5 + x^2 + 1$
8	$x^8 + x^6 + x^5 + x + 1$
9	$x^9 + x^4 + 1$
10	$x^{10} + x^3 + 1$

[#]Polynomial that requires minimum number of Ex-OR gates for given degree.

$x^{5-5} + x^{5-2} + x^{5-0}$
i.e. $x^5 + x^3 + 1$
is also primitive polynomial

Internal Feedback



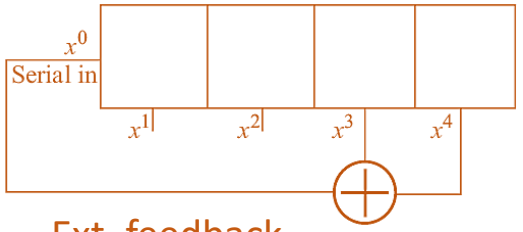
$f(x) = x^4 + x^3 + 1$

Cycle length = 15

Pseudorandom sequence with int. feedback:
101011001000111..

Pseudorandom sequence with ext. feedback:
000100110101111.. (earlier)

Q	*R*	*S*	*T*	*Serial in = T*	*Input to T FF*	*Clock cycle*
1	1	1	1	1	0	1
1	1	1	0	0	1	2
0	1	1	1	1	0	3
1	0	1	0	0	1	4
0	1	0	1	1	1	5
1	0	1	1	1	0	6
1	1	0	0	0	0	7
0	1	1	0	0	1	8
0	0	1	1	1	0	9
1	0	0	0	0	0	10
0	1	0	0	0	0	11
0	0	1	0	0	1	12
0	0	0	1	1	1	13
1	0	0	1	1	1	14
1	1	0	1	1	1	15
1	1	1	1	1	0	(repeats)

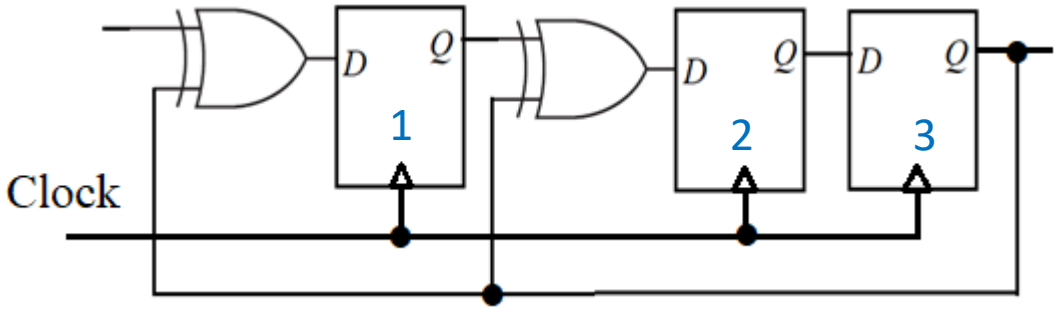


Ext. feedback

Primitive polynomial for external feedback also gives maximal length for internal feedback and generates pseudorandom sequence (different).

External Input

Example: Cycle Redundancy Check (CRC)



$g(x) = x^3 + x + 1$

Transmitter

Message: **1100101(000)**
Remainder: 010
Coded message: **1100101010**

3 check bits

Receiver

Remainder: 000
No error
CRC is specially useful for detecting burst error

Clock	S_{in}	Q_1	Q_2	Q_3
0	**1**	0	0	0
1	**1**	1	0	0
2	**0**	1	1	0
3	**0**	0	1	1
4	**1**	1	1	1
5	**0**	0	0	1
6	**1**	1	1	0
7	**0**	1	1	1
8	**0**	1	0	1
9	**0**	1	0	0
10	-	**0**	**1**	**0**

$Q_1Q_2Q_3$: initialized with 000

Transmitter

Clock	S_{in}	Q_1	Q_2	Q_3
0	**1**	0	0	0
1	**1**	1	0	0
2	**0**	1	1	0
3	**0**	0	1	1
4	**1**	1	1	1
5	**0**	0	0	1
6	**1**	1	1	0
7	**0**	1	1	1
8	**1**	1	0	1
9	**0**	0	0	0
10	end	**0**	**0**	**0**

Receiver

$D_1 = S_{in} \oplus Q_3$

$D_2 = Q_1 \oplus Q_3$

$D_3 = Q_2$

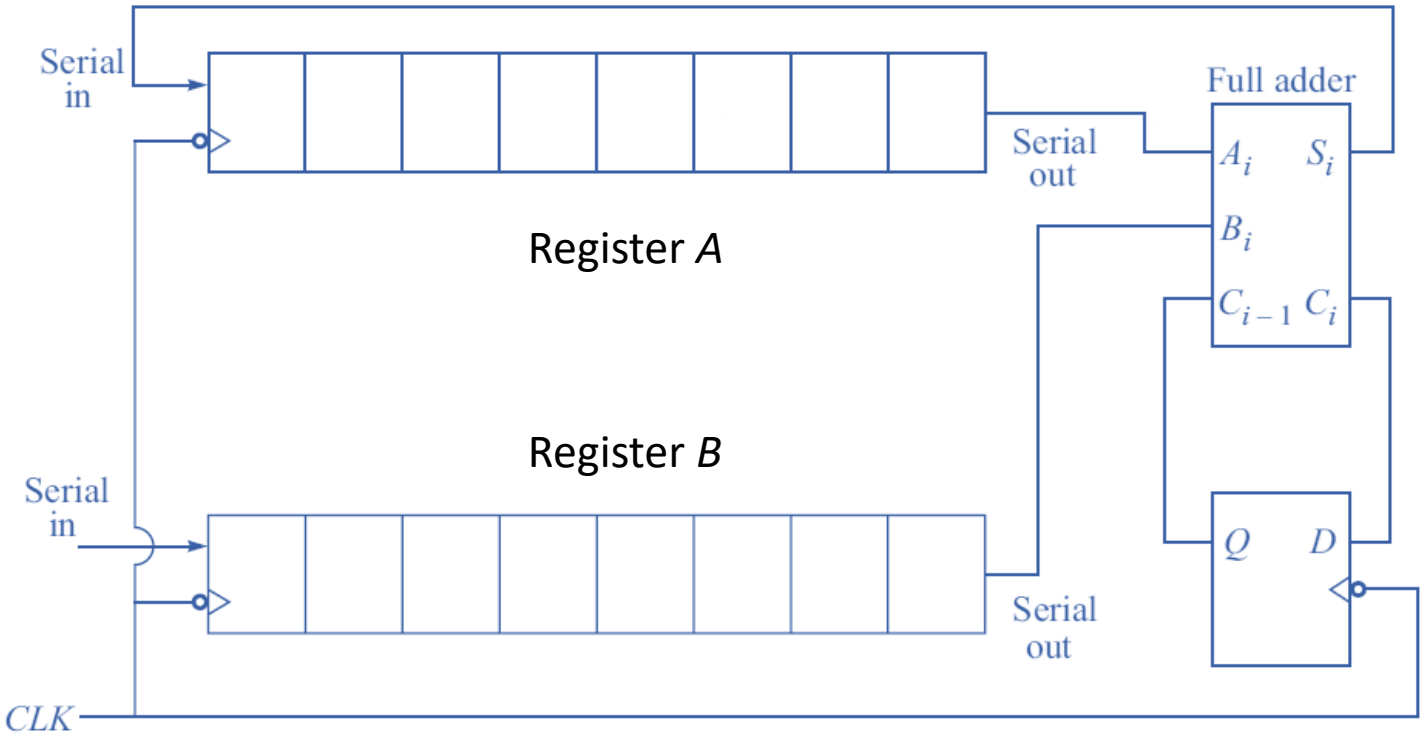
$\boldsymbol{Q_{n+1} = D_n}$

Application

- Fast Counter: Simpler feedback for which higher clock rate is possible.
- Test pattern generator: Pseudorandom pattern is efficient in high fault-coverage of Application-Specific Integrated Circuit (ASIC).
- Scrambling: LFSR output is Ex-Ored with data to widen the bandwidth.
- Cryptography: Pseudorandom numbers are generated from an LFSR with a seed value which serves as cryptographic key and provides efficient encryption / decryption.
- Error Control Code: Used in Cycle Redundancy Check (CRC) for data transmission and storage. It is popular as it is easy to implement.

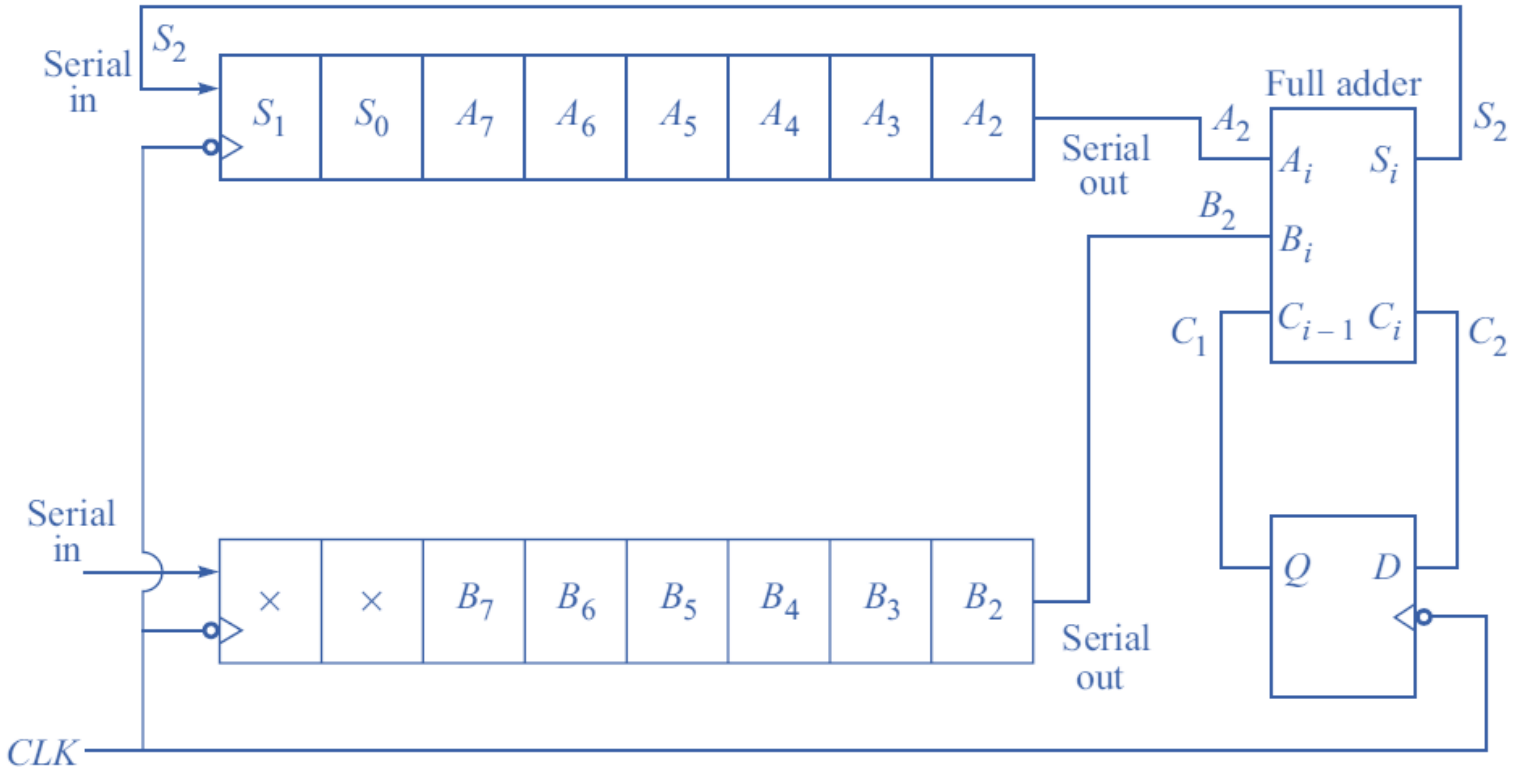
  (CRC-16: $g(x) = x^{16} + x^{15} + x^2 + 1$ can detect up to 16 burst error)

Serial Addition



- Initially, register *A* and *B* store the addend and augend.
- Least significant bits are first out serially when addition starts.
- Initially, *D* flip-flop is reset. It stores the carry generated from *i*-th bit addition and feed that as input to (*i*+1)-th bit addition.
- Sum bit is serially entered in *A.*
- If required, next number to be added with former two can be serially entered in *B*.

Serial Addition



8 clock cycles to complete 8 bit addition

Clock 0: *A*: 00001010 *B*: 00001011 *Q*: 0

Clock 1: *A*: 10000101 *B*: x0000101 *Q*: 0

Clock 2: *A*: 01000010 *B*: xx000010 *Q*: 1

…

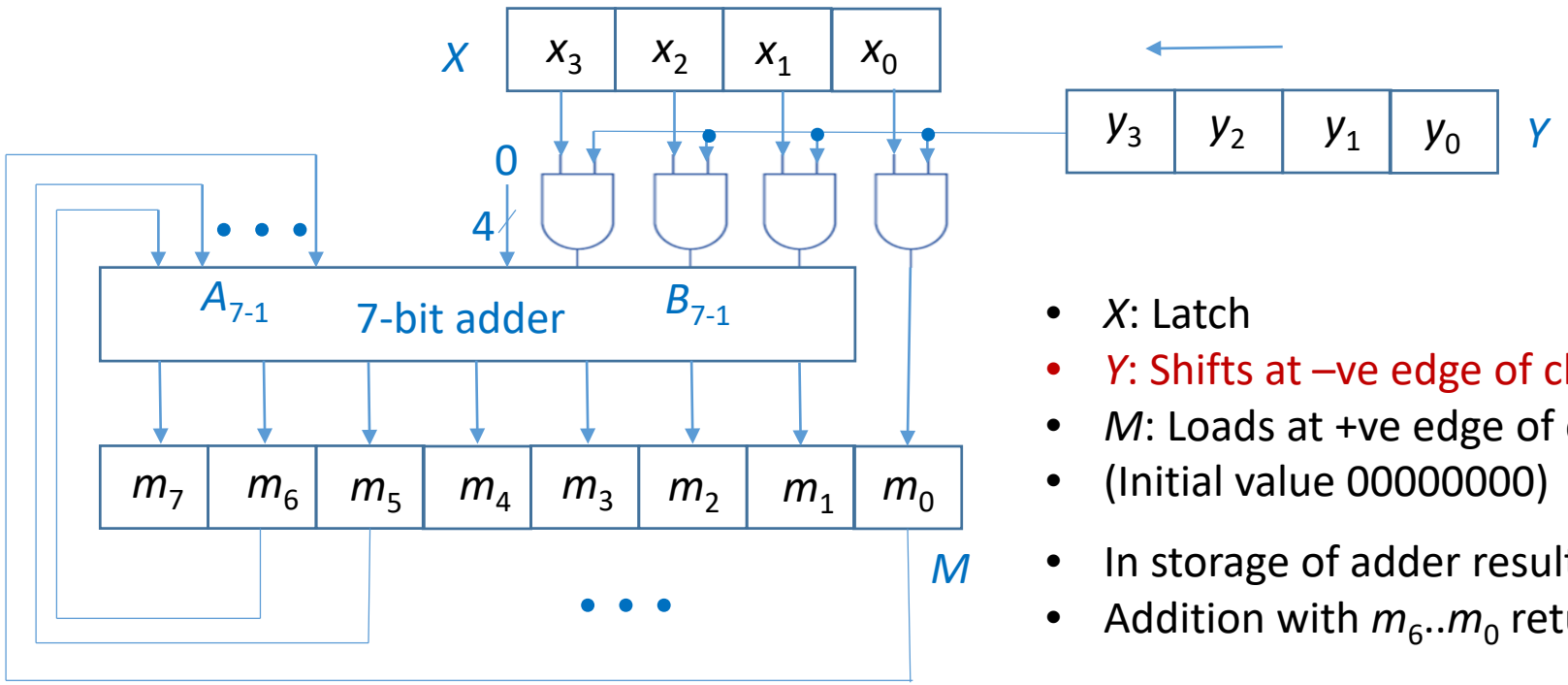
…

x: 0 or 1 depending on next no., if any, to be added

Serial Multiplication

$x_3x_2x_1x_0$: 1101
$y_3y_2y_1y_0$: 1011

1101	$(13)_{10}$
1011	$(11)_{10}$
---	---
10001111	$(143)_{10}$

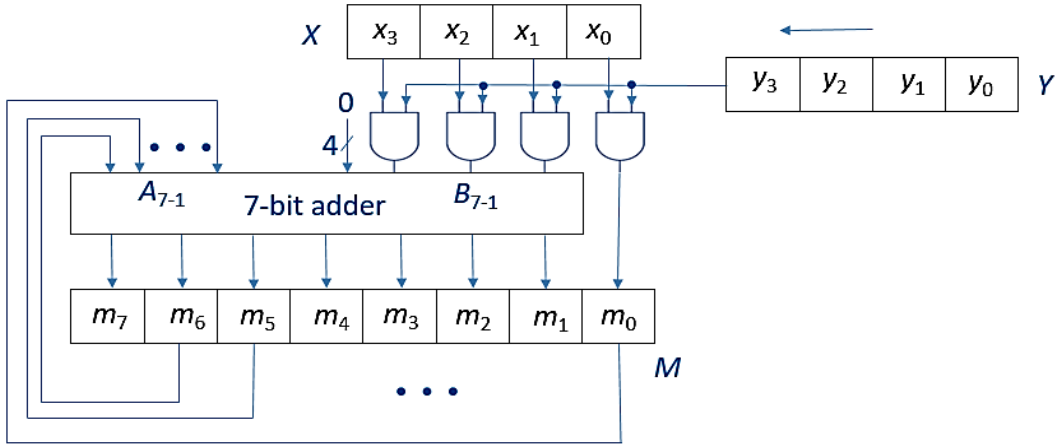


- *X*: Latch
- *Y*: Shifts at –ve edge of clock
- *M*: Loads at +ve edge of clock
- (Initial value 00000000)

- In storage of adder result, one left shift
- Addition with $m_6..m_0$ returns as $m_7..m_1$

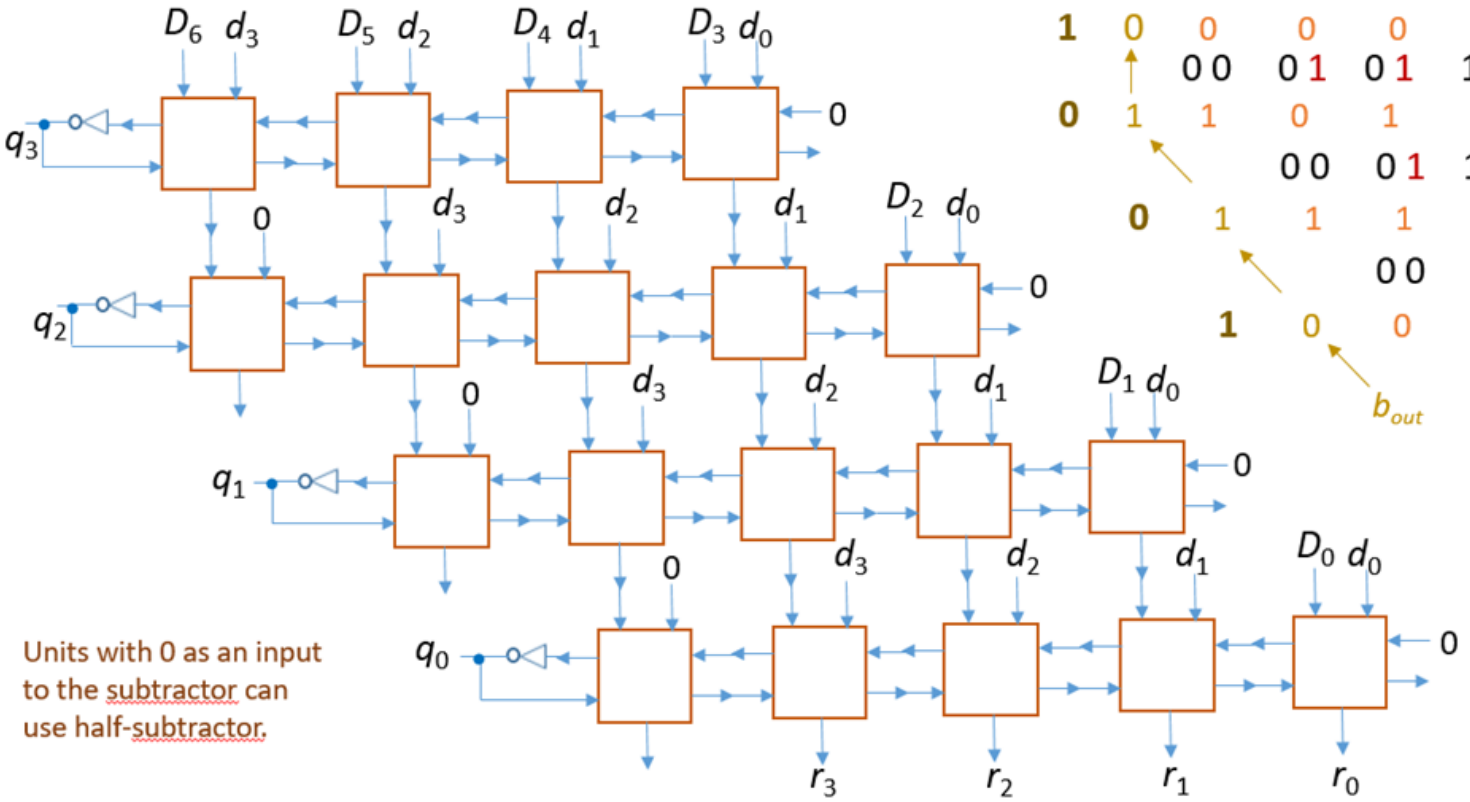
Example

$x_3x_2x_1x_0$: 1101
$y_3y_2y_1y_0$: 1011

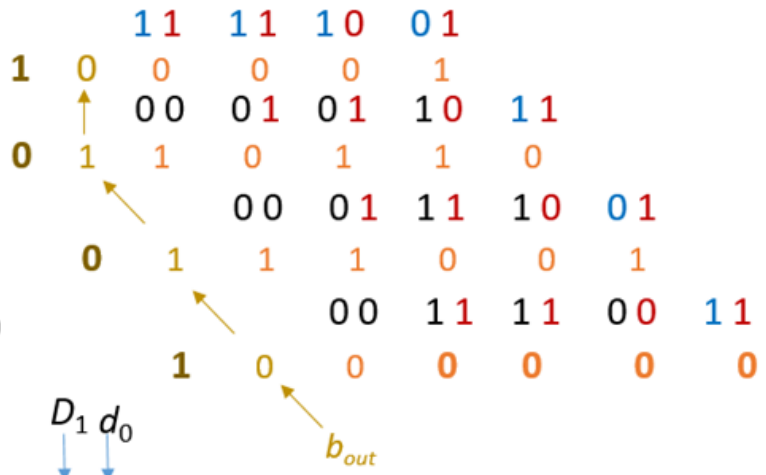


$x_3x_2x_1x_0$:	1101	1101		
y_3 : / y_2 :	1	0	• • •	
AND o/p:	1101	0000	1101	1101
M output:	00000000	00001101	00011010	01000001
M at adder i/p (in blue):	00000000	00011010	00110100	10000010
Other adder i/p (in blue):	00001101	00000000	00001101	00001101
----------	----------	----------	----------	----------
Addition result (in blue):	00001101	00011010	01000001	10001111
M at clock trigger:	00001101	00011010	01000001	**10001111**
Clock	1	2	3	4

Review: Combinatorial Divider Circuit

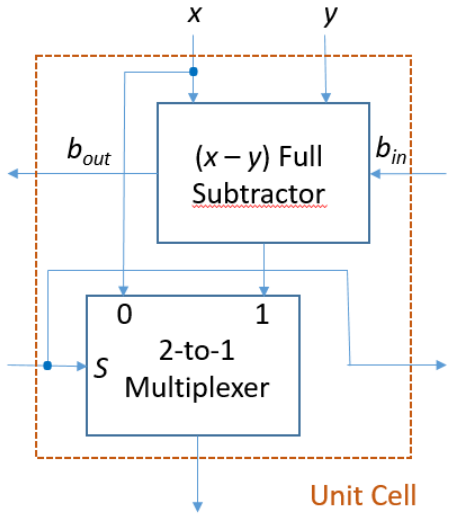


Units with 0 as an input to the subtractor can use half-subtractor.

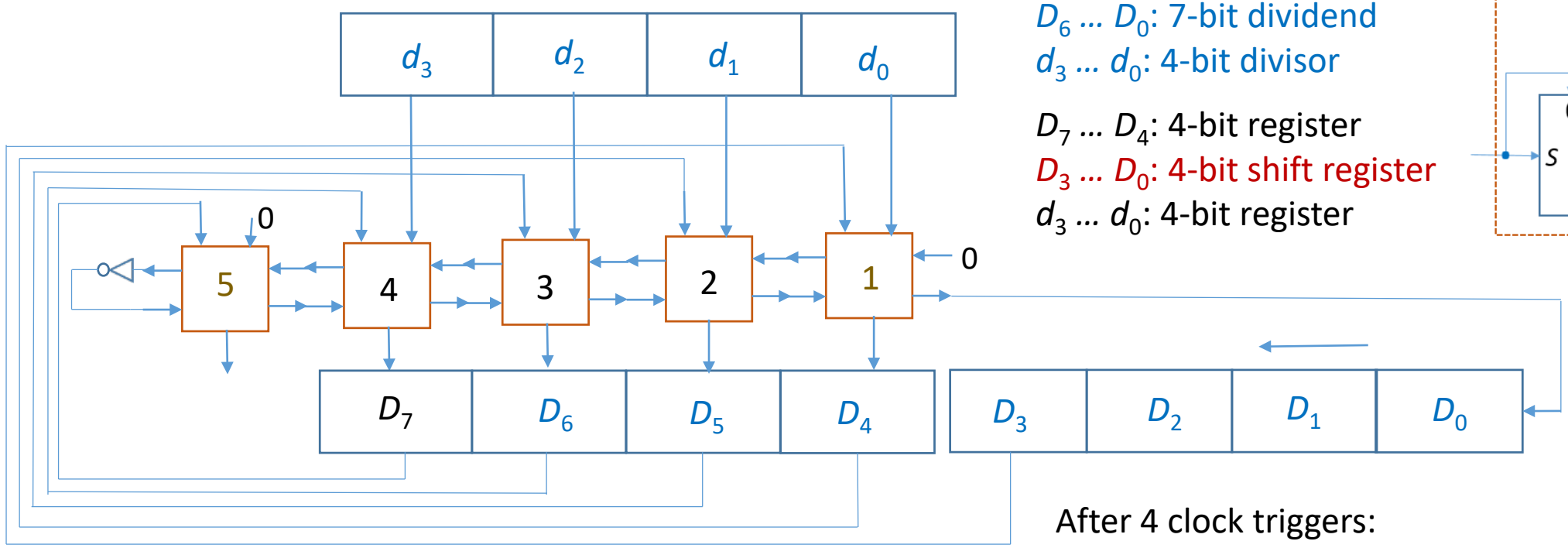


D: 1110101
d: 1101

q: 1001
r: 0000

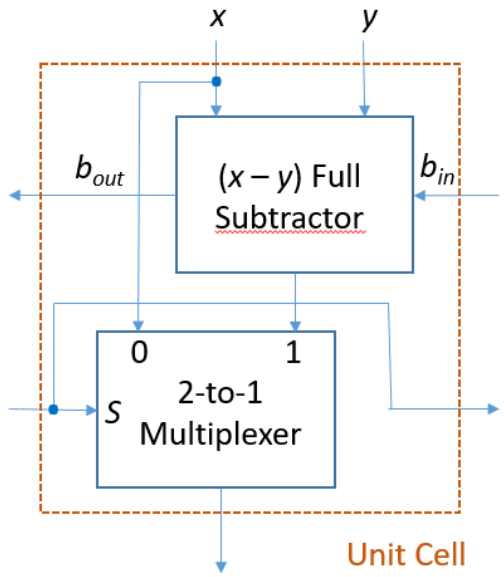


Serial Division



$D_6 \ldots D_0$: 7-bit dividend
$d_3 \ldots d_0$: 4-bit divisor

$D_7 \ldots D_4$: 4-bit register
$D_3 \ldots D_0$: 4-bit shift register
$d_3 \ldots d_0$: 4-bit register



1 and 5 can use
Half-subtractor

After 4 clock triggers:

$D_7 \ldots D_4$: 4-bit remainder
$D_3 \ldots D_0$: 4-bit quotient

References:

- Donald P. Leach, Albert P. Malvino, and Goutam Saha, Digital Principles & Applications 8e, McGraw Hill
- Lloris Ruiz A., Castillo Morales E., Parrilla Roure L., García Ríos A. Number Systems. In: Algebraic Circuits. Intelligent Systems Reference Library, vol 66. Springer, Berlin, Heidelberg