CE233(CAD), Lecture 3:

Xilinx FPGAs

Mehdi Modarressi

Department of Electrical and Computer Engineering,

University of Tehran

- Reading:
  - *Spartan-3 Generation FPGA User Guide, Xilinx ug331 (v1.8)*
  - “FPGA Prototyping By VHDL Examples- Xilinx Spartan-3 Version”, Chapter 2
  - “The Design Warrior's Guide to FPGAs”, Chapter 4

Xilinx FPGAs

Xilinx modern FPGAs

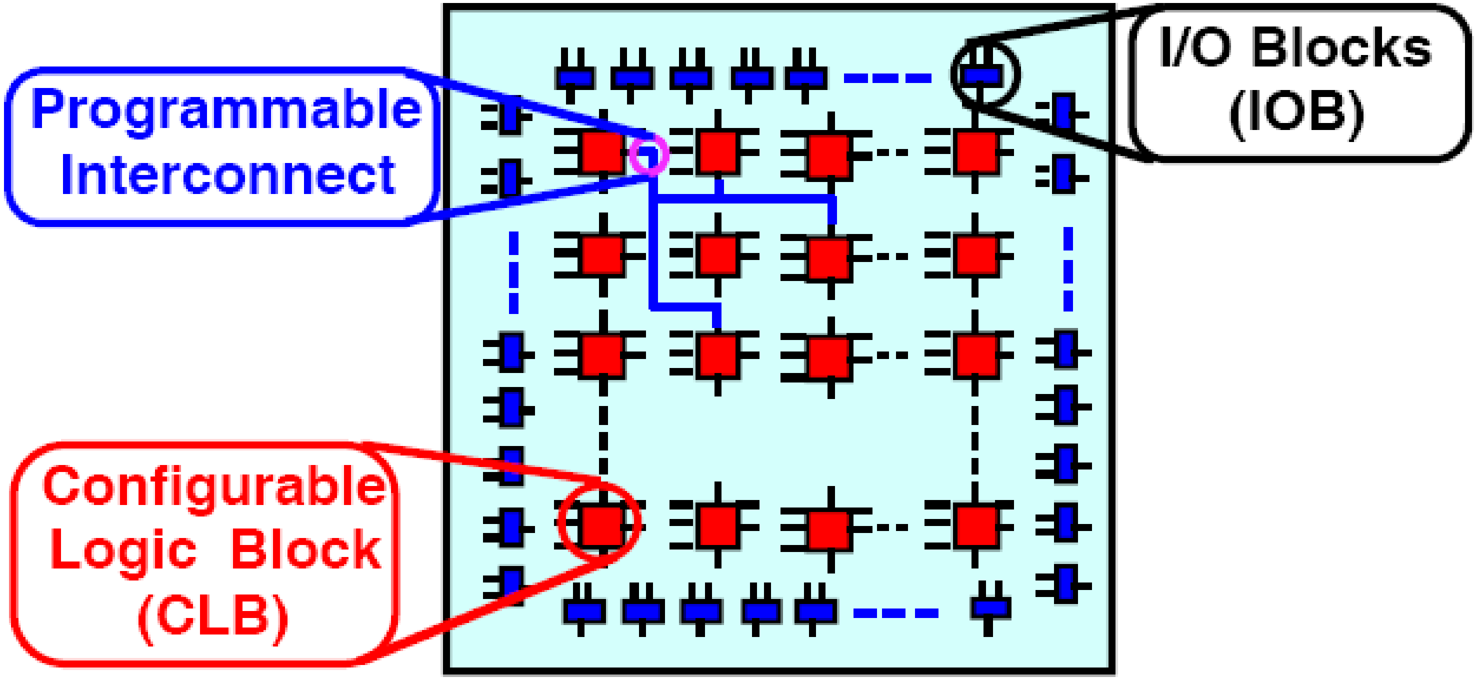
Xilinx FPGA History

08 Spartan-3A XA
'08 Virtex-5 FXT
'08 Virtex-5 TXT
'07 QPro Virtex-4
'07 QPro Virtex-IIPro
'07 Spartan-3AN
'07 Spartan-3A DSP
'06 Virtex-5 LX/LXT/SXT
'06 Spartan-3A
'05 Spartan-3E
'04 Virtex-4 LX/SX/FX
'04 Spartan-3 XA
'04 Spartan-IIE XA
'04 QPro-R Virtex-II
'04 QPro Virtex-II
'04 QPro Virtex-E
'04 Spartan-3L
'03 Virtex-II ProX
'03 Spartan-3
'02 Virtex-II Pro
'02 Spartan-IIE IQ
'02 Spartan-II IQ
'02 SpartanXL IQ
'01 Spartan-IIE
'01 QPro-R Virtex
'00 Virtex-II

'99 Virtex-E
'99 Spartan-II
'99 QPro-Virtex
'98 Virtex
'98 SpartanXL
'98 Spartan
'98 XQ4000XL
'98 XQR4000XL
'97 XC4000XV
'97 XC4000XLA
'97 XC4000XL
'97 XQ4000E/EX
'96 XC4000E/EX
'96 XC6200
'95 XC8100
'95 XC5200
'94 XC4000D/L
'93 XC4000H
'93 XC3100A/L
'93 XC3100
'93 XC3000A/L
'93 XC2000L
'92 XC4000A
'91 XC4000
'87 XC3000
'85 XC2000

Example: Xilinx XC4000

- A simple FPGA introduced in the 80’s
- Array of *Configurable Logic Blocks* (*CLBs*)
- SRAM-based



Source: lecture 10 by Hassan Bhatti

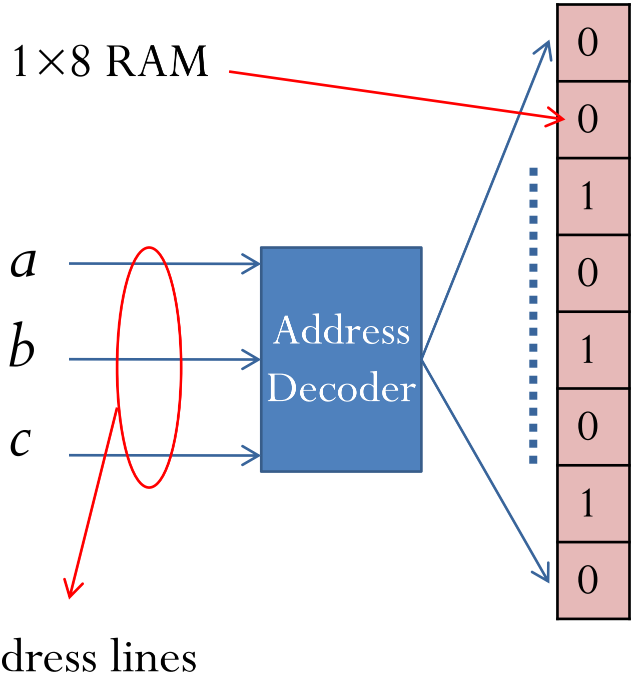
Lookup table (LUT)

- Each XC4000 CLB implements the logic functions using a lookup table (LUT)
  - An alternative to PAL-based implementation
- An n-input LUT can implement any n-input function

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Truth table

$$F=(a+b)\times c'$$

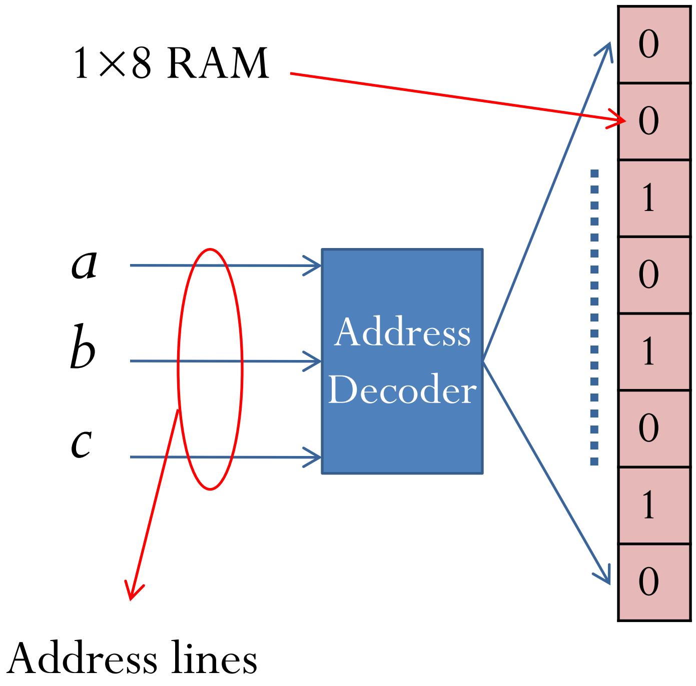


LUT vs PAL

$$F=(a+b)\times c'$$

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Truth table



1×8 RAM

a

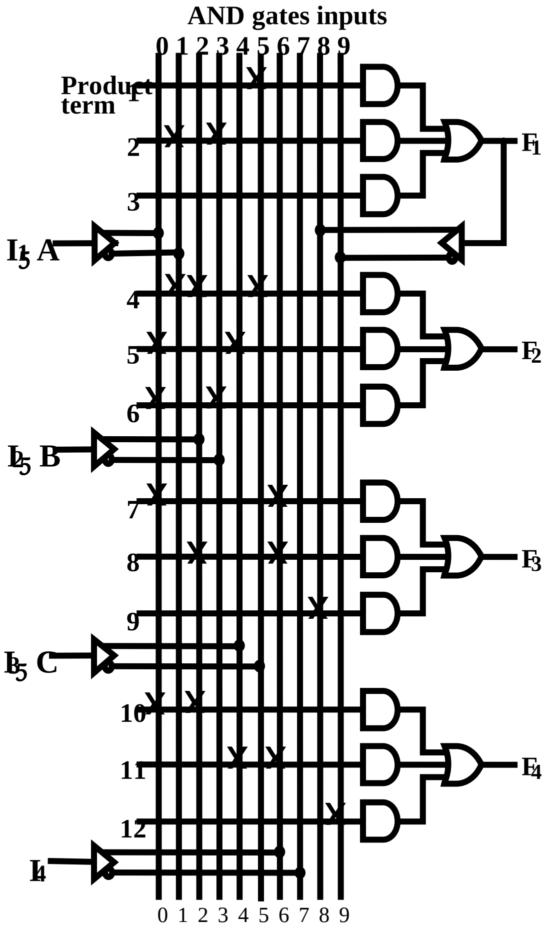
b

c

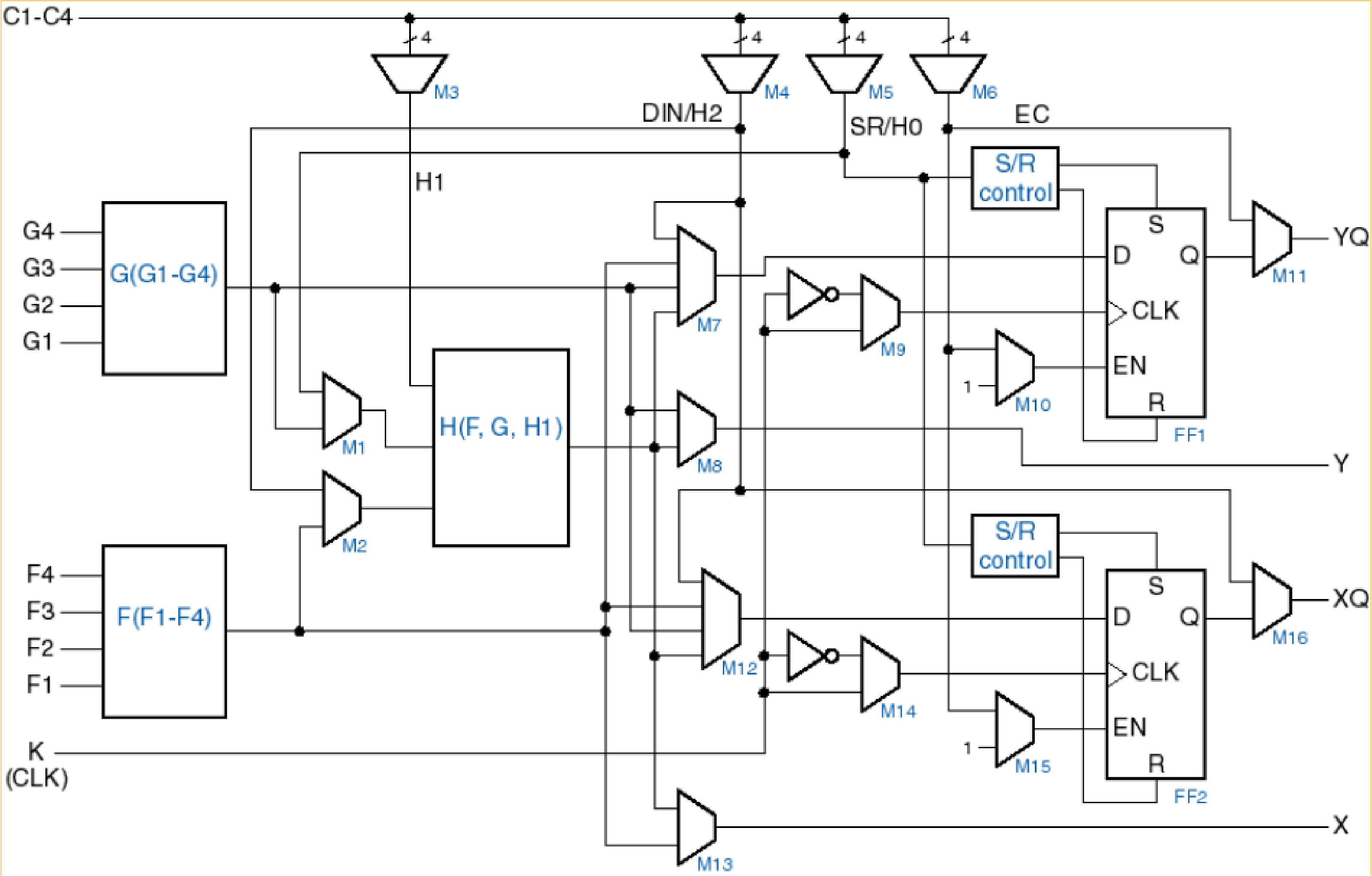
Address Decoder

0
0
1
0
1
0
1
0

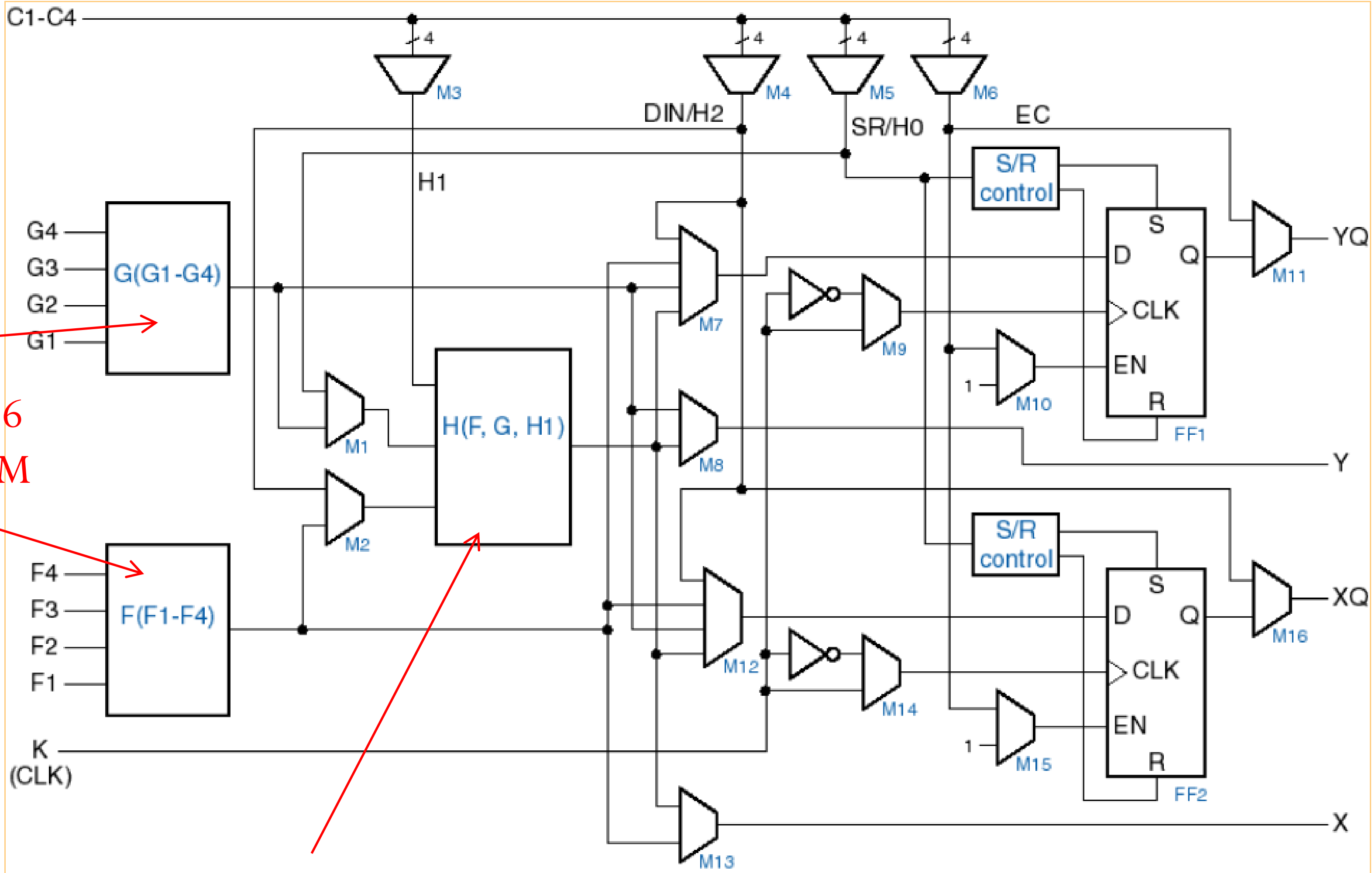
Address lines



Xilinx XC4000 FPGAs- CLB



Xilinx XC4000 FPGAs- CLB

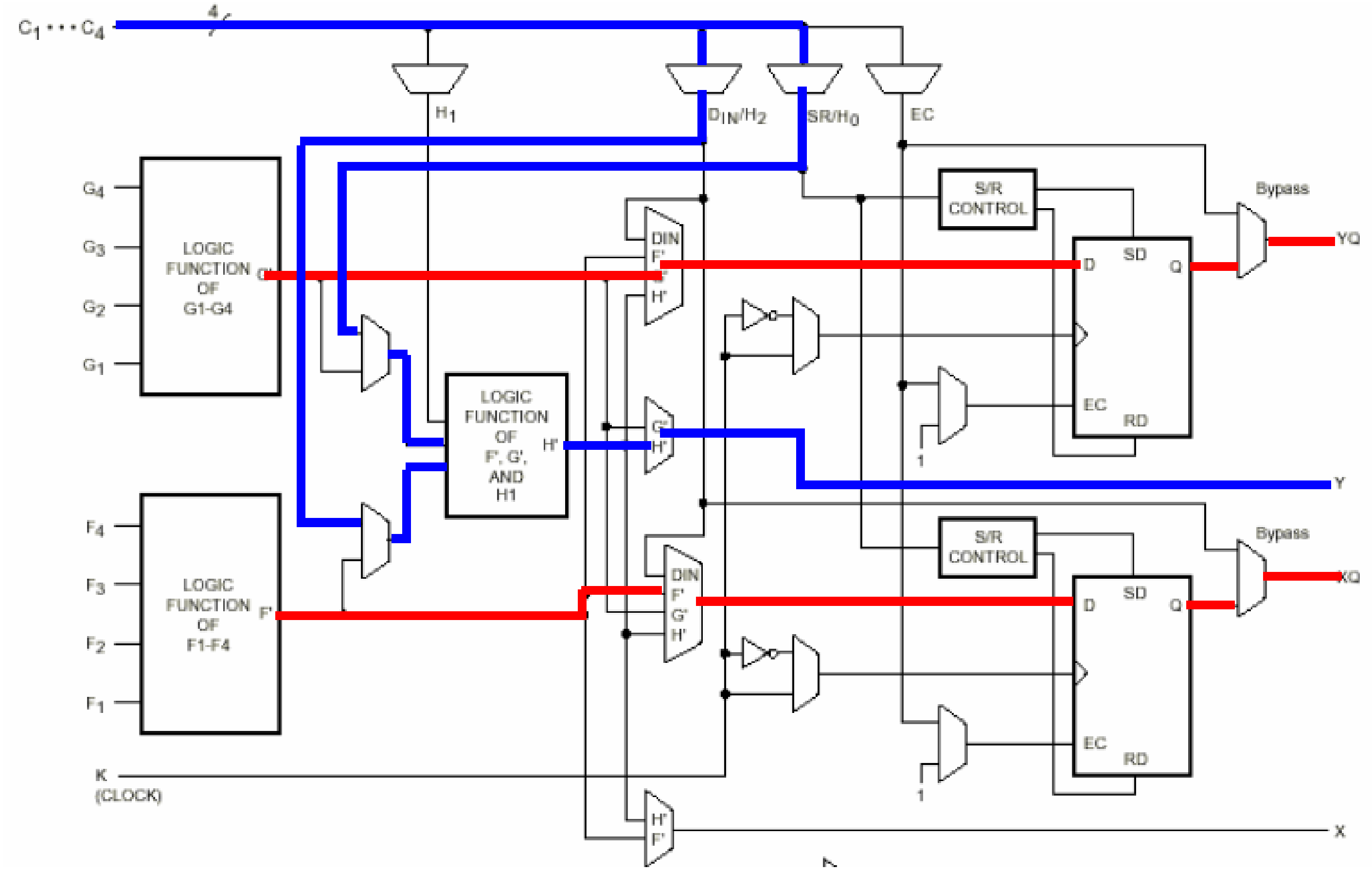


F and G LUTs: 4 input -> 16 bits of RAM (each)

H LUT : 3 input –> 8 bits of RAM

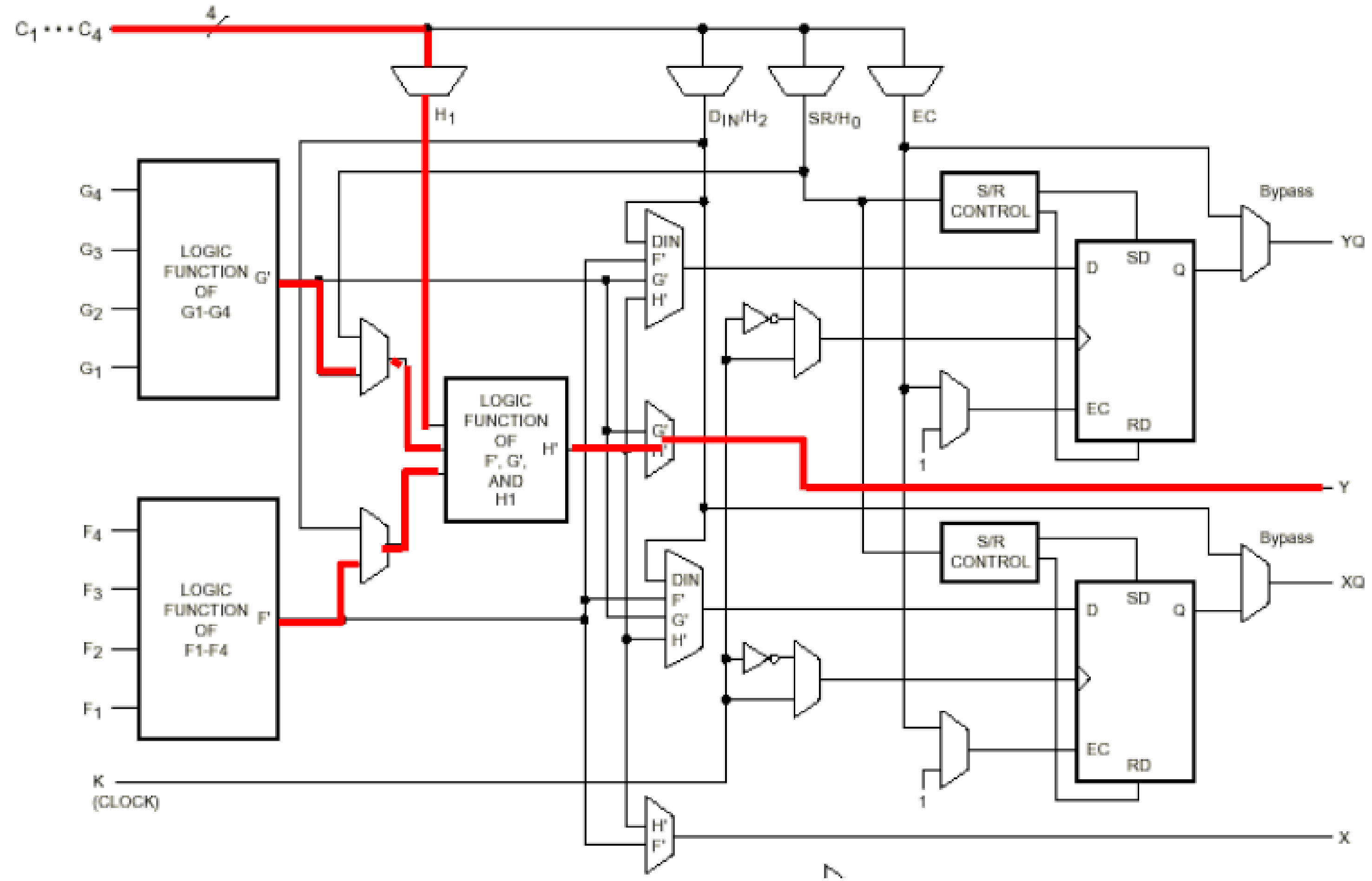
Xilinx XC4000 CLB configuration

- Two 4-input functions, registered output input functions, and one 2-input function with combinational output



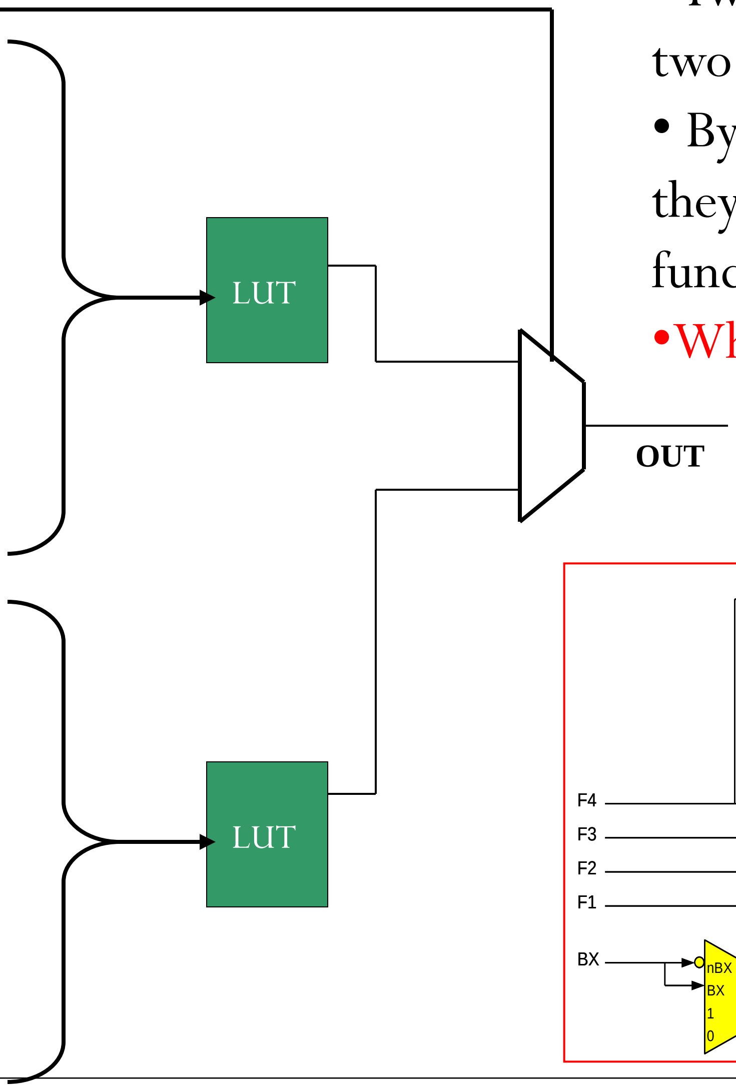
Xilinx C4000 CLB configuration

- 5-input function, combinational output

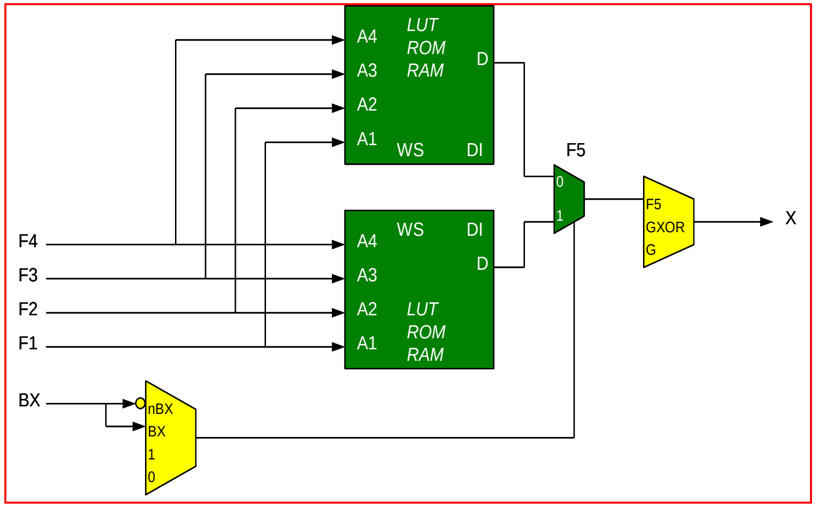


(X+1)-Input functions by x-input LUTs

X_5	X_4	X_3	X_2	X_1	Y
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	0



- Two 4-inpu LUTs can implement two 4-input functions
- By adding a multiplexer like below, they can implement any 5-input functions
- Why?



(X+1)-Input functions by x-input LUTs

- Two n-input LUTs and one 3-input LUT can implement any (n+1)-input function
- Why?
- Prove as an exercise!

Example

- Implement the following functions on a single CLB of the XC4000 FPGA:

$$X = A'B' (C + D)$$

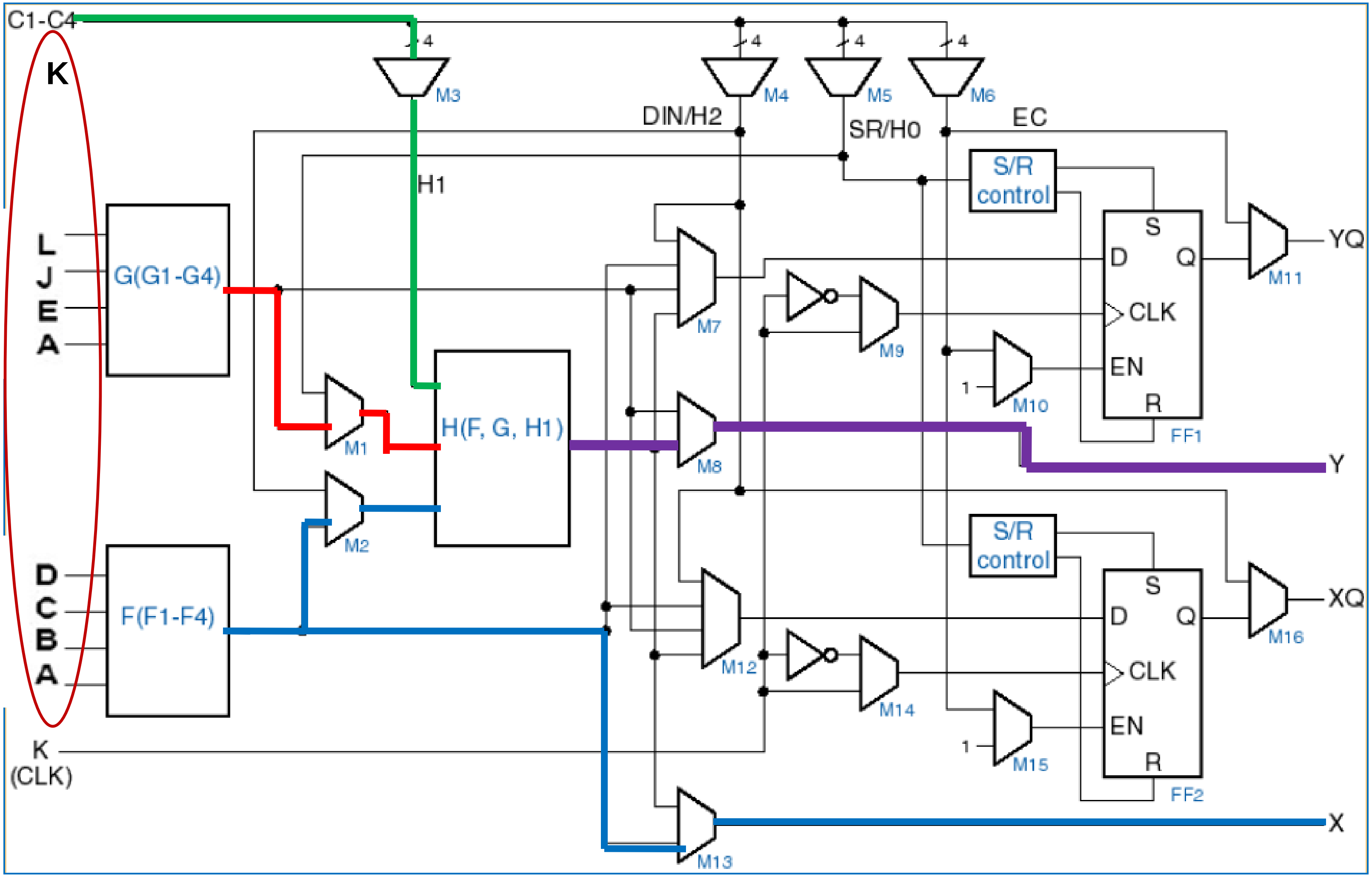
$$Y = AK + BK + C'D'K + AEJL$$

- Use look up table F to implement X
- Use look up table G for AEJL
- Use F, G and H for Y:

$$Y = K(A+B + C'D') + AEJL$$

$$= KX' + AEJL= KF'+G$$

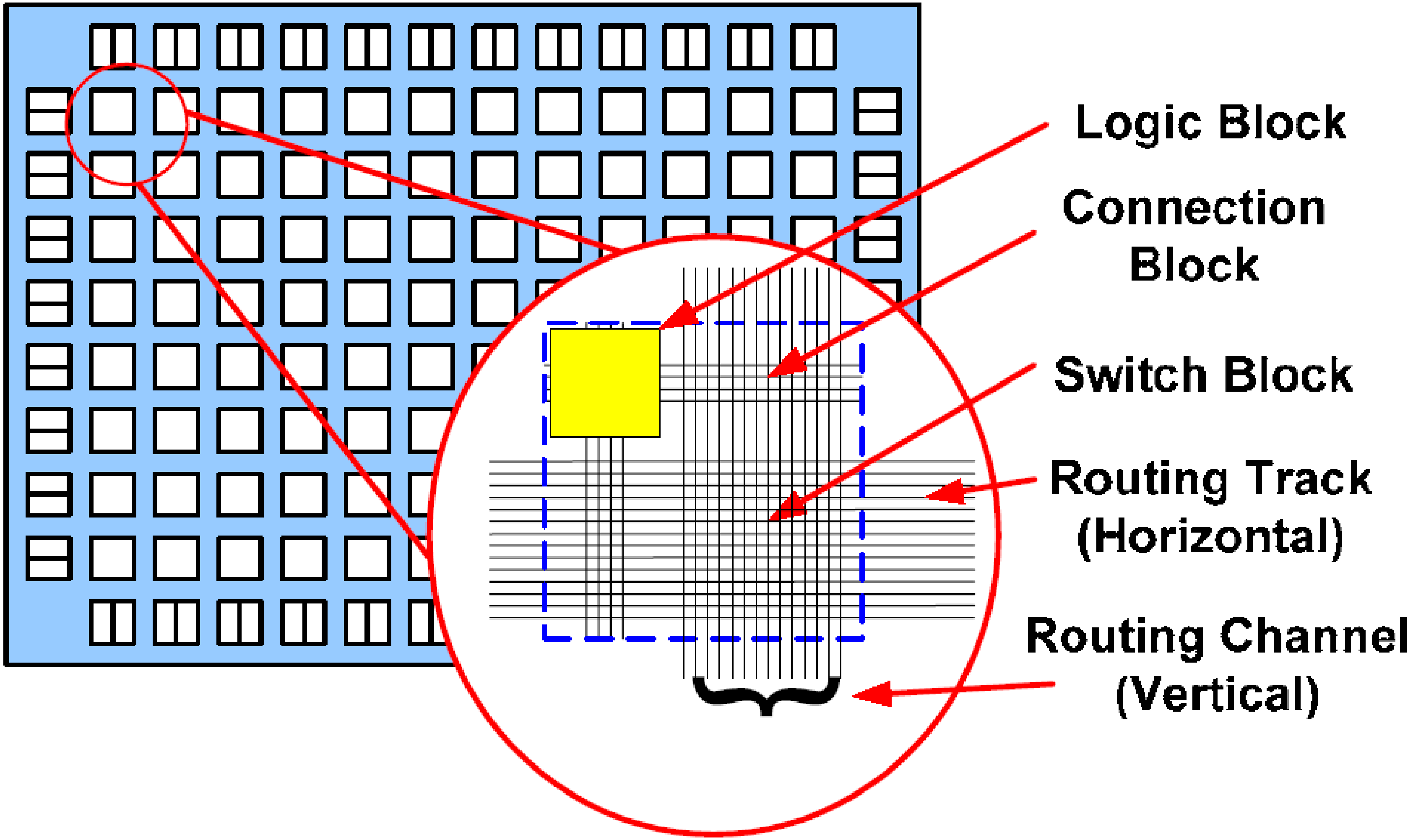
Programming a CLB



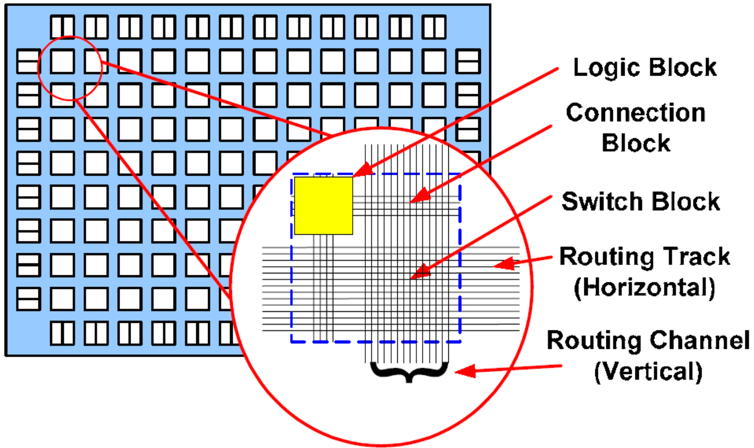
$$X = A'B'(C + D)$$

$$Y = AK + BK + C'D'K + AEJL$$

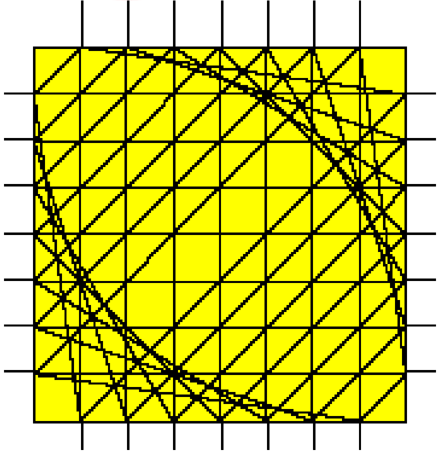
FPGA interconnects



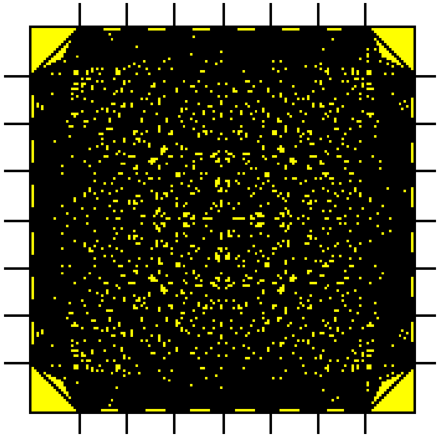
FPGA interconnects- switch box



- Switch Blocks connect horizontal and vertical channels
- A trade-off between flexibility and switch size

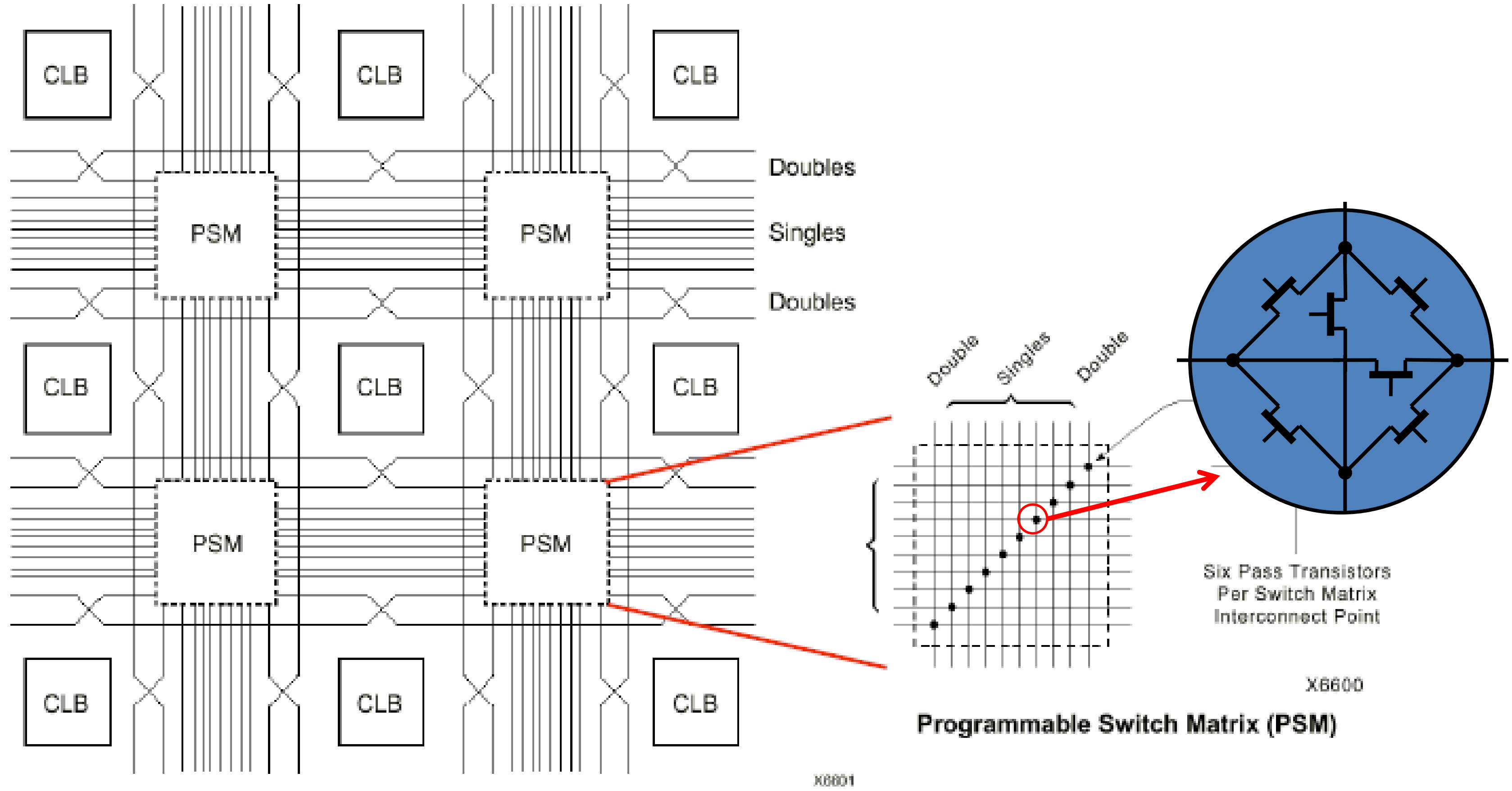


Fs = 3, the most common switch
Low flexibility, good size



Fs = n-1=6: too big but very flexible

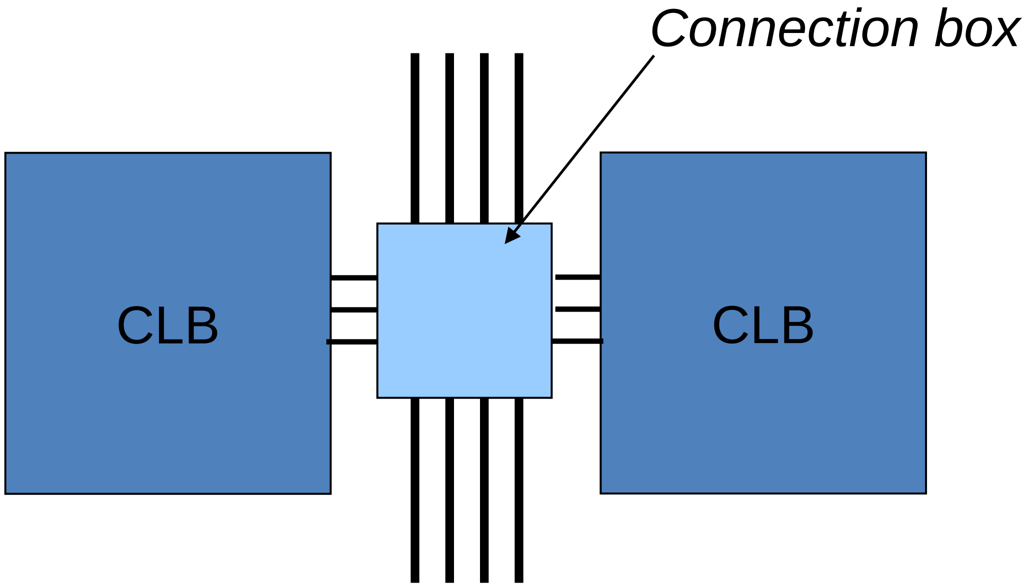
XC4000 interconnects



Fs=3, why?

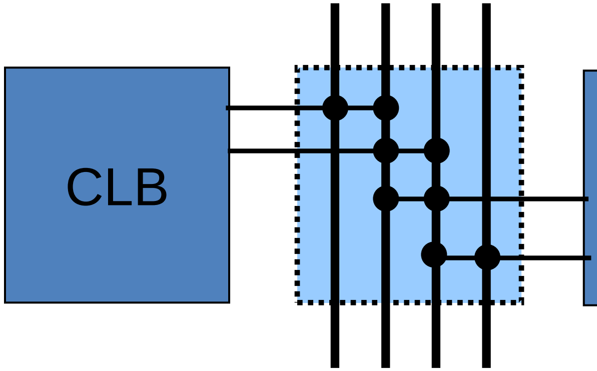
Connection box

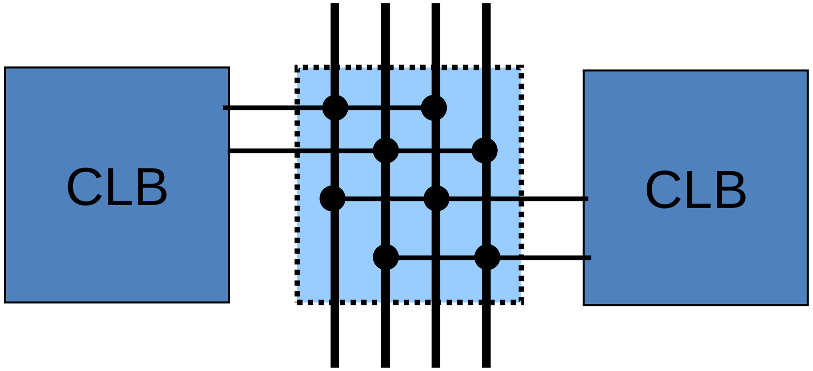
- How to connect CLB to wires?
- Solution: Connection box
  - Device that allows inputs and outputs of CLB to connect to different wires



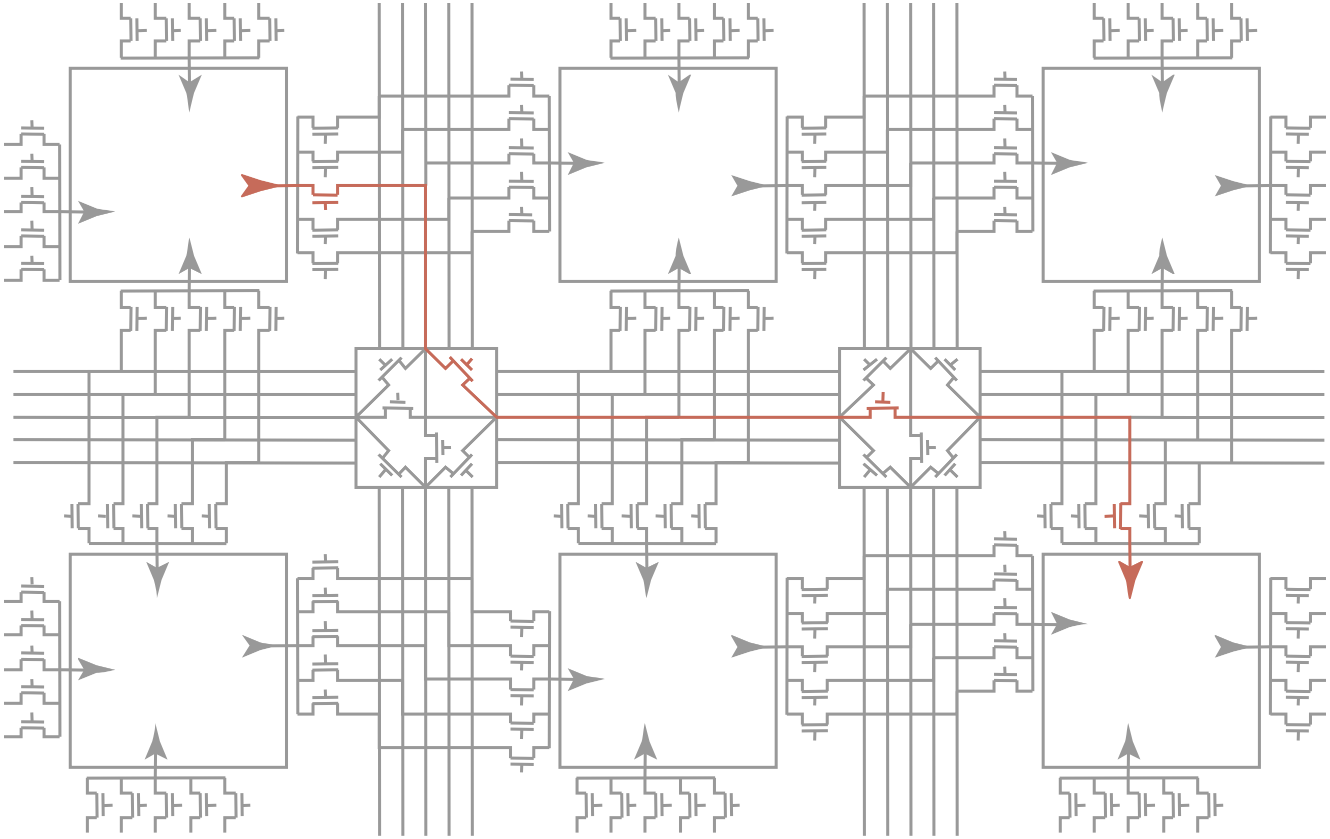
Connection box

- Connection box characteristics
  - Topology
    - Defines the specific wires each CLB I/O can connect to
    - Examples: same flexibility, different topology





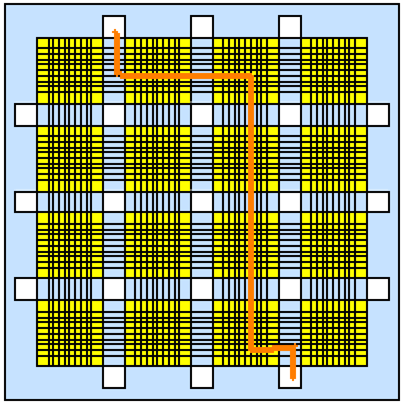
FPGA interconnects



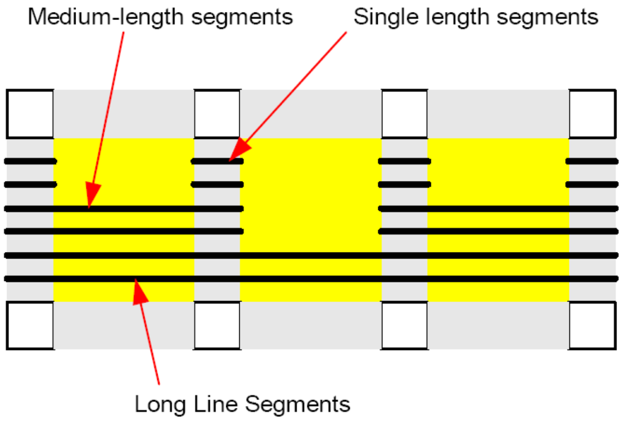
Connect Logic Blocks using wires and Programmable Switches

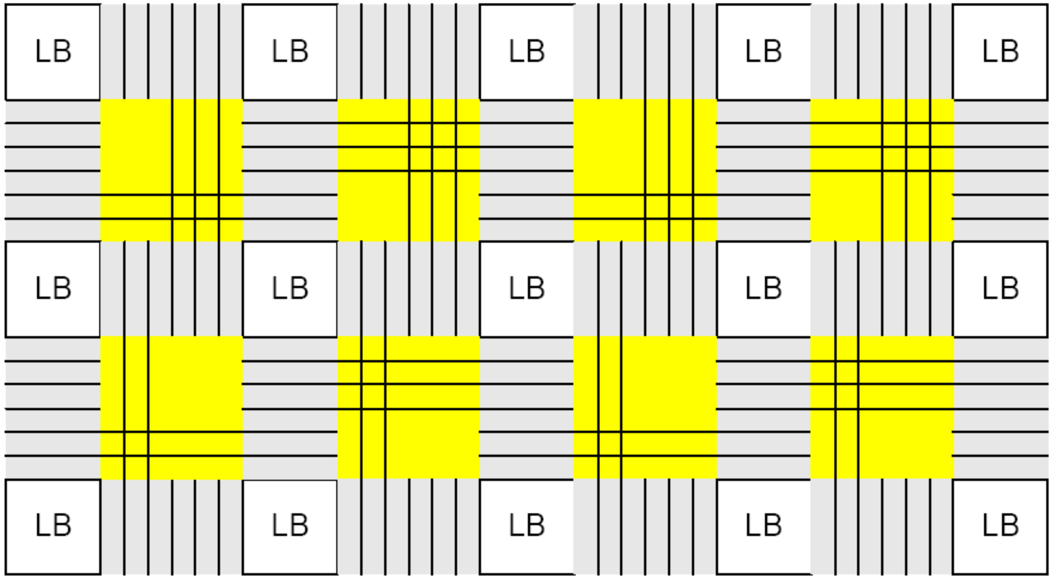
FPGA interconnects- wires

- Typically, an FPGA contains a mix of segment lengths:
  - Some wires that span only one logic block
  - Some wires that span more than one logic block
  - Some wires that span the whole chip
- If a segment is too short, must traverse many segments to reach your destination
  - If a segment is too long, waste routing capacity, extra capacitance: too slow!



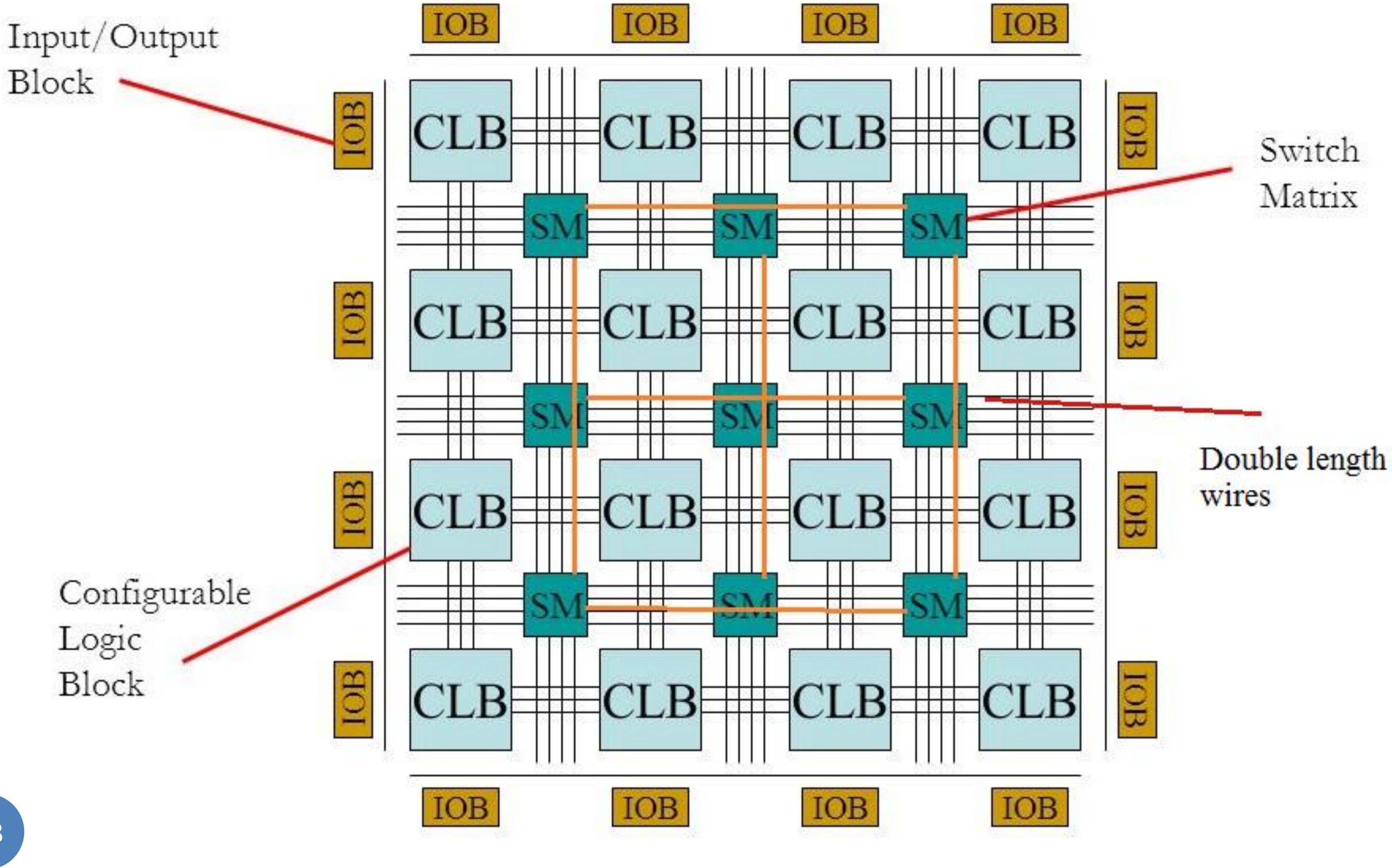
FPGA interconnects- wires





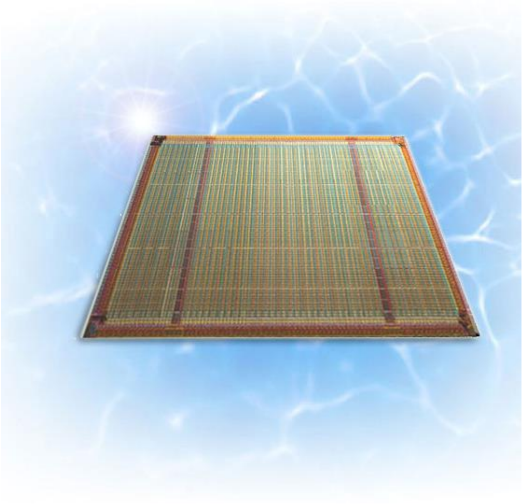
- Short segments are good for local connections
- Long segments are good for global connections
- Most FPGA's have a variety of segment lengths

Long wires


Input/Output Block
Switch Matrix
Double length wires
Configurable Logic Block
IOB
CLB
SM

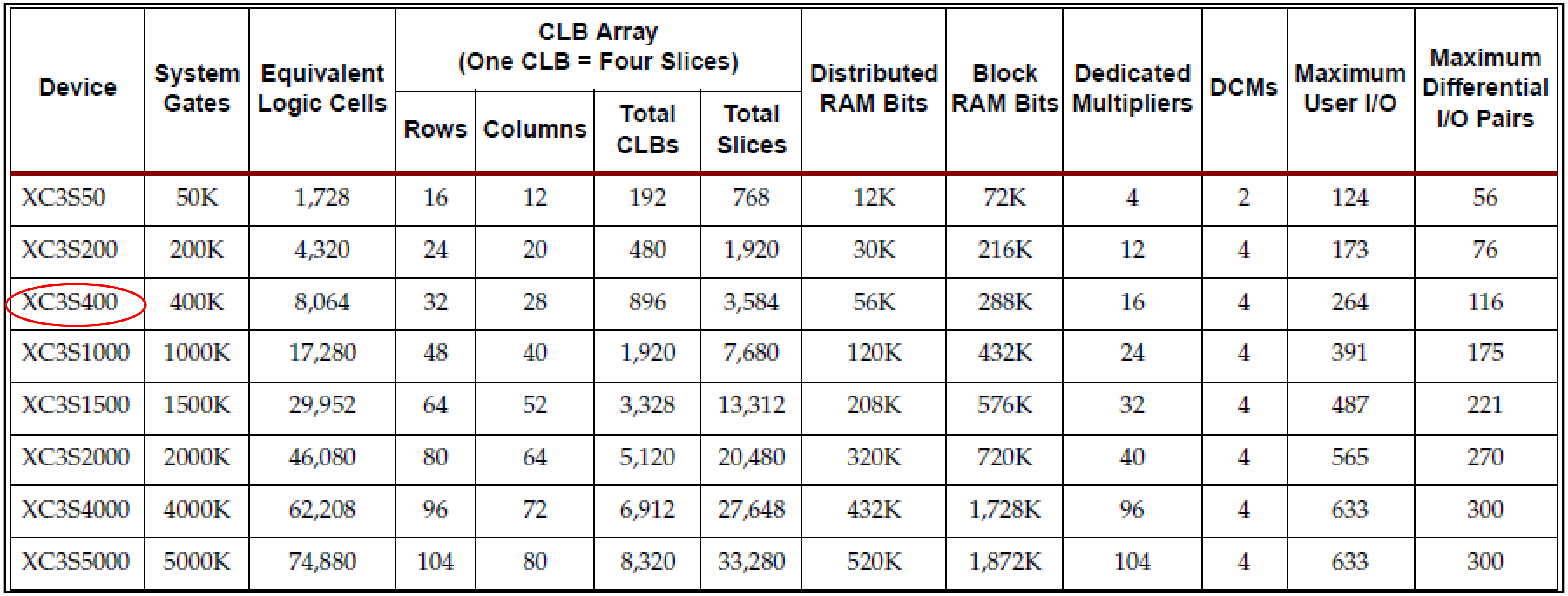
Spartan-3 FPGAs

- Spartan: Cost-effective FPGA family of Xilinx
  - Spartan-6 is the last generation of the family
  - Cheaper and smaller than Virtex (the high-performance FPGAs of Xilinx)
- The industry's first 90 nm FPGA
- Different generations: Spartan-3, Spartan-3E, Spartan-3A



Spartan-3

- In the CAD lab, you will work with Spartan-3 XCS400 FPGAs

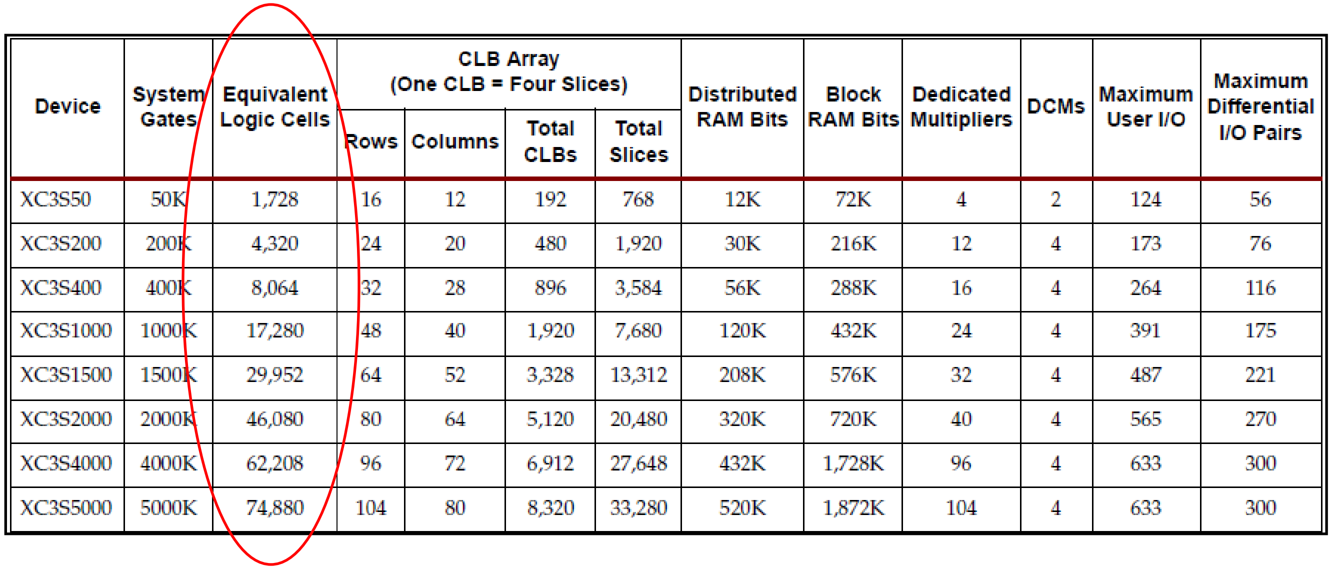


Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM Bits	Block RAM Bits	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S50	50K	1,728	16	12	192	768	12K	72K	4	2	124	56
XC3S200	200K	4,320	24	20	480	1,920	30K	216K	12	4	173	76
XC3S400	400K	8,064	32	28	896	3,584	56K	288K	16	4	264	116
XC3S1000	1000K	17,280	48	40	1,920	7,680	120K	432K	24	4	391	175
XC3S1500	1500K	29,952	64	52	3,328	13,312	208K	576K	32	4	487	221
XC3S2000	2000K	46,080	80	64	5,120	20,480	320K	720K	40	4	565	270
XC3S4000	4000K	62,208	96	72	6,912	27,648	432K	1,728K	96	4	633	300
XC3S5000	5000K	74,880	104	80	8,320	33,280	520K	1,872K	104	4	633	300

Source: Spartan-3 Generation FPGA User Guide, Xilinx ug331 (v1.8)

Logic cell

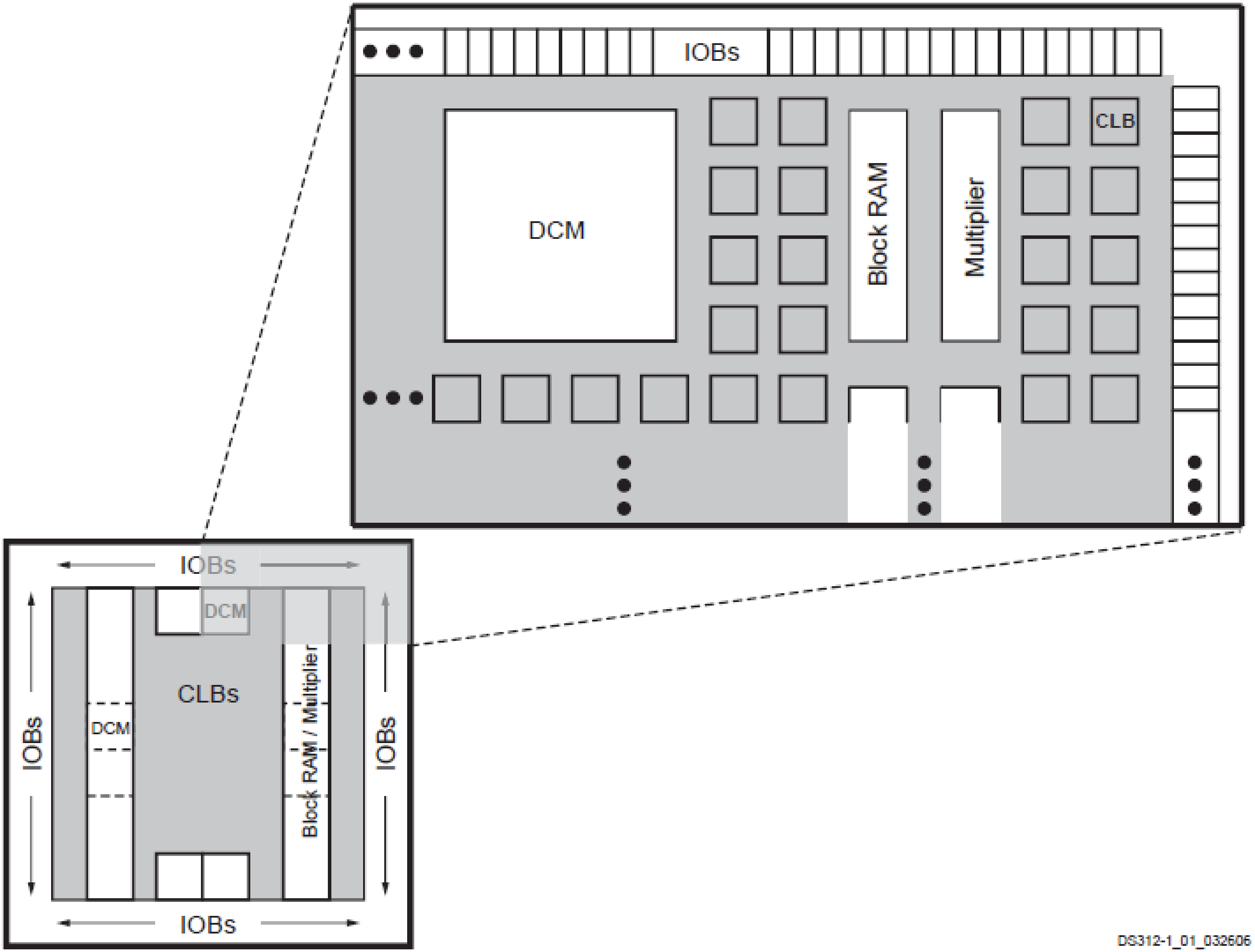
- Logic cell: LUT + Flip-flop
- The additional features in a slice, such as the wide multiplexers, carry logic, and arithmetic gates, add to the capacity of a slice
  - Implementing logic that would otherwise require additional LUTs
- Benchmarks show that the overall slice is equivalent to 2.25 simple logic cells



Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM Bits	Block RAM Bits	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S50	50K	1,728	16	12	192	768	12K	72K	4	2	124	56
XC3S200	200K	4,320	24	20	480	1,920	30K	216K	12	4	173	76
XC3S400	400K	8,064	32	28	896	3,584	56K	288K	16	4	264	116
XC3S1000	1000K	17,280	48	40	1,920	7,680	120K	432K	24	4	391	175
XC3S1500	1500K	29,952	64	52	3,328	13,312	208K	576K	32	4	487	221
XC3S2000	2000K	46,080	80	64	5,120	20,480	320K	720K	40	4	565	270
XC3S4000	4000K	62,208	96	72	6,912	27,648	432K	1,728K	96	4	633	300
XC3S5000	5000K	74,880	104	80	8,320	33,280	520K	1,872K	104	4	633	300

Source: Spartan-3 Generation FPGA User Guide, Xilinx ug331 (v1.8)

Spartan-3 architecture



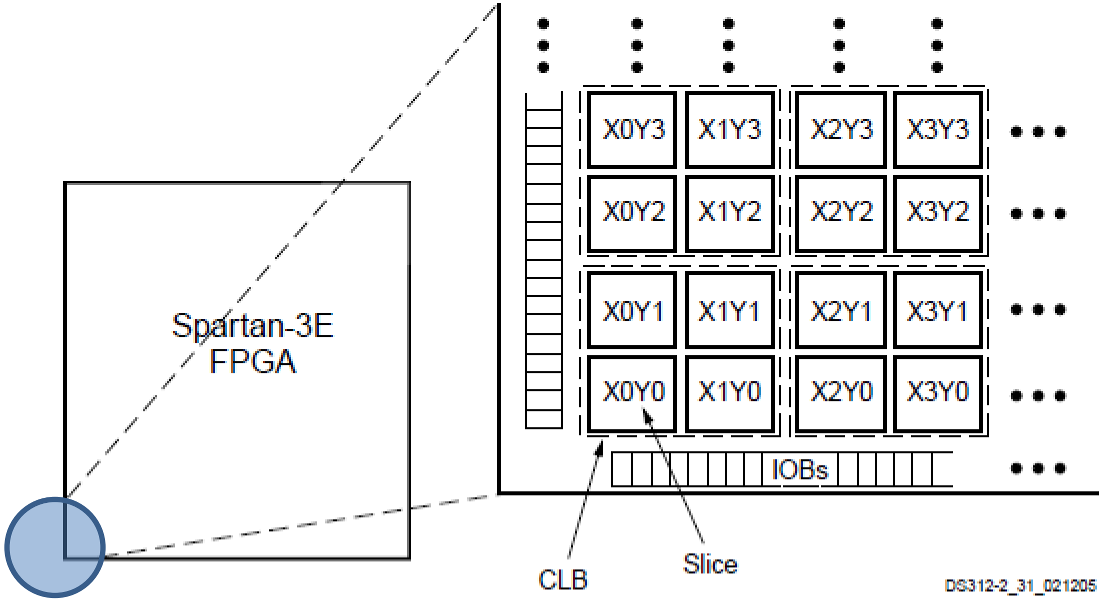
Source: Spartan-3 Generation FPGA User Guide, Xilinx ug331 (v1.8)

Spartan-3 components

- Configurable Logic Blocks (CLB)
- Flexible Synchronous Memory (BlockRAM)
- Programmable Input Output Blocks (IOB)
- Dedicated multipliers

Configurable Logic Blocks (CLB)

- Main logic resource for implementing synchronous as well as combinatorial circuits
- Arranged in a regular array of rows and columns

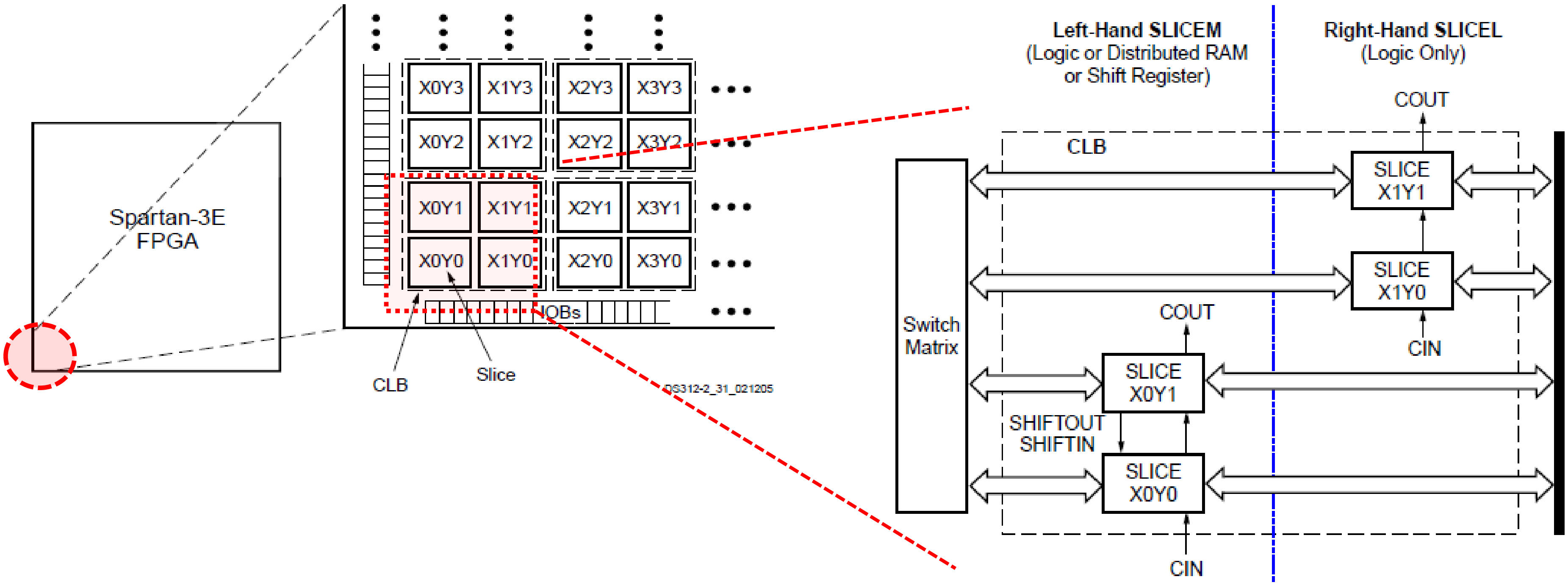


X0Y3	X1Y3	X2Y3	X3Y3	• • •
X0Y2	X1Y2	X2Y2	X3Y2	• • •
X0Y1	X1Y1	X2Y1	X3Y1	• • •
X0Y0	X1Y0	X2Y0	X3Y0	• • •

Source: Spartan-3 Generation FPGA User Guide, Xilinx ug331 (v1.8)

CLB and slice

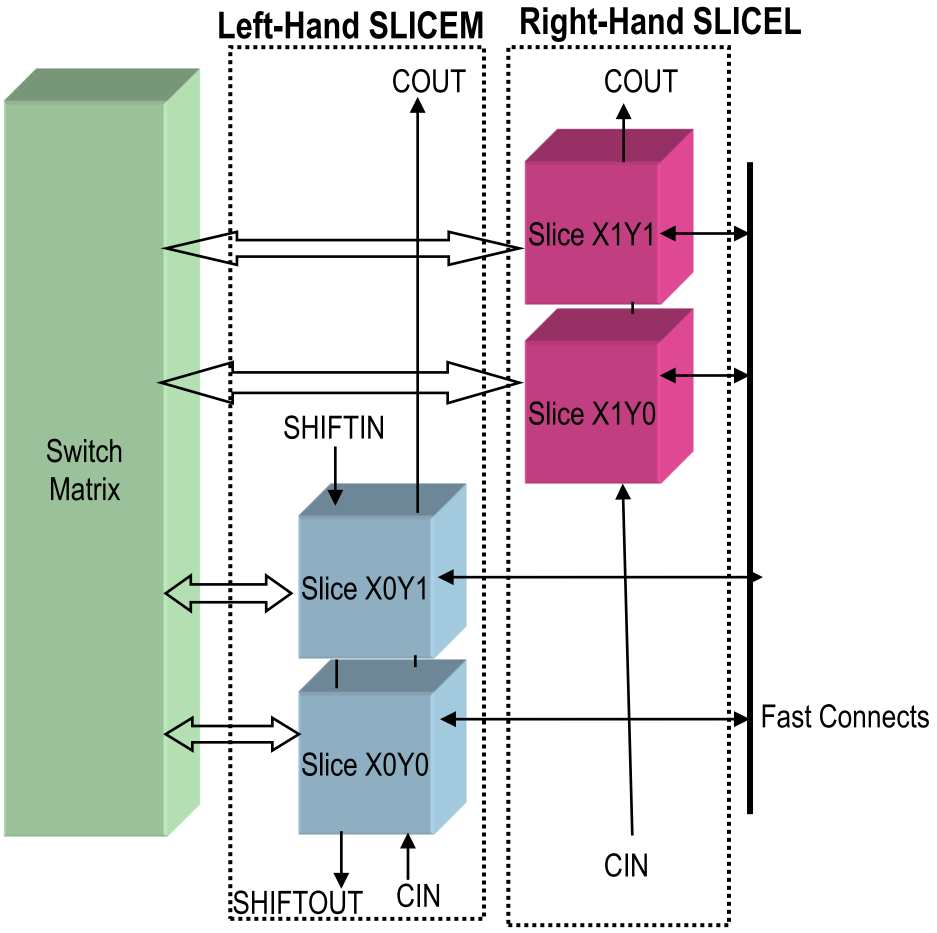
- Each Spartan-3 CLB contains four slices
- Fast interconnects among the slices of a CLB



Source: Spartan-3 Generation FPGA User Guide, Xilinx ug331 (v1.8)

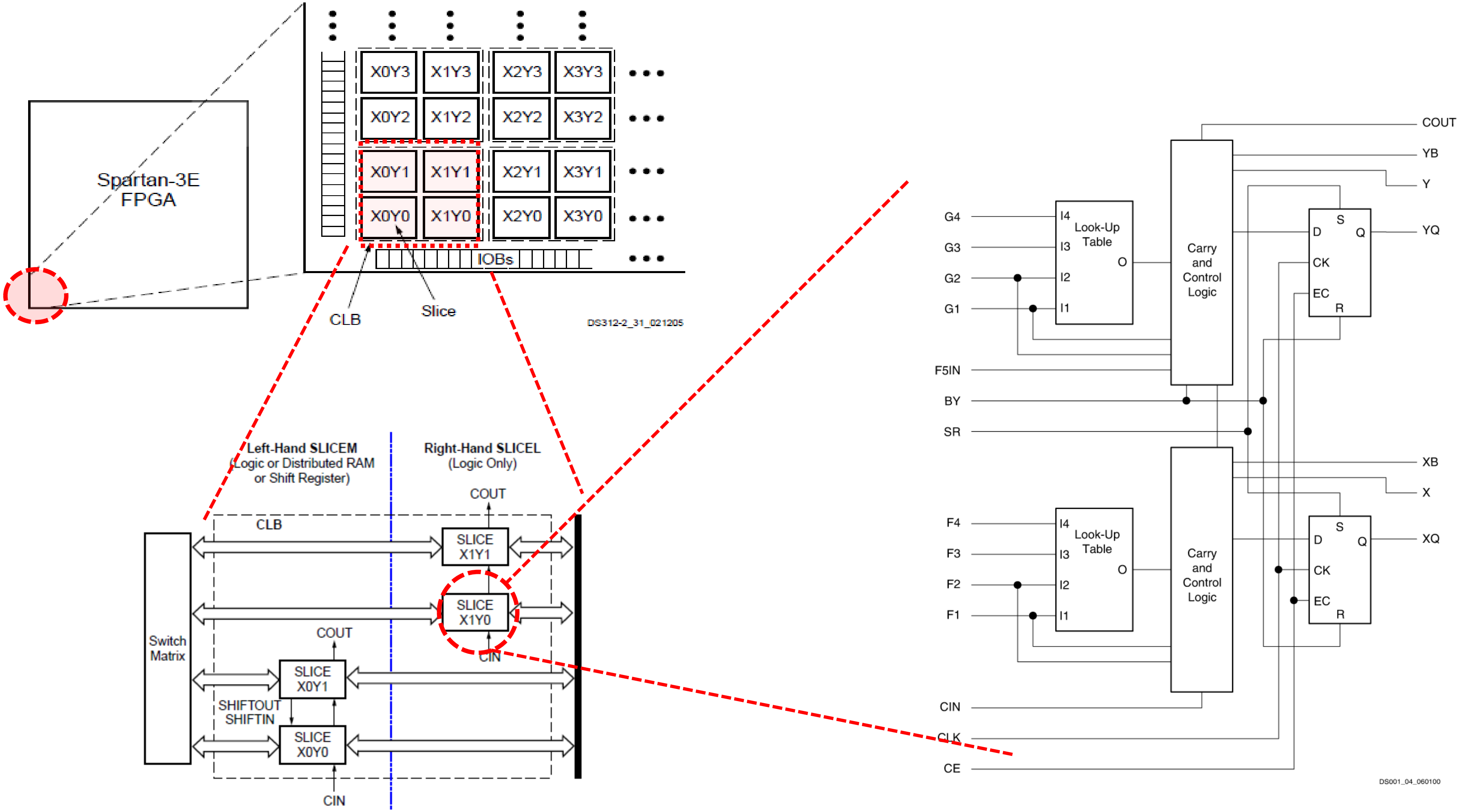
SLICEM and SLICEL

- Slices are grouped in pairs in two columns
- Left-hand SLICEM (Memory)
  - In addition to logic implementation, LUTs can be configured as memory or 16-bit shift register
- Right-hand SLICEL (Logic)
  - LUTs can be used as logic only



Source: Basic FPGA Architecture, Xilinx university program

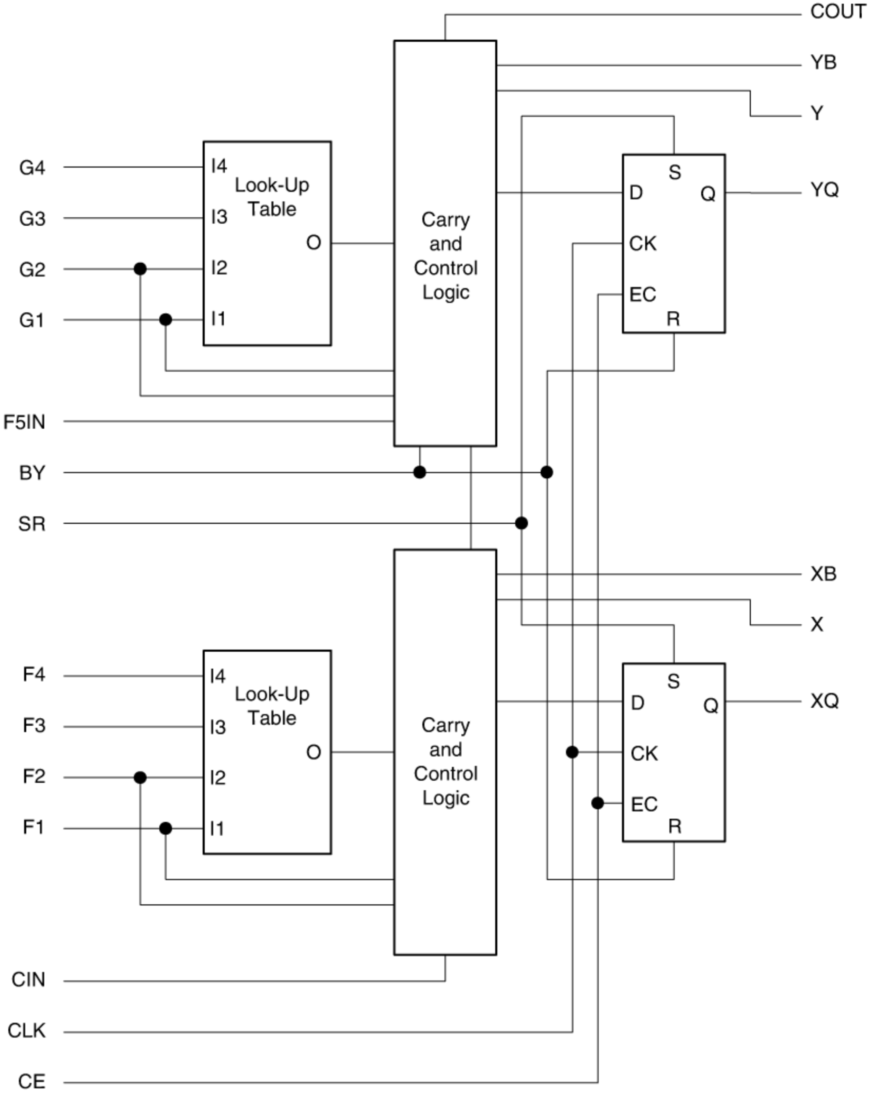
CLB and slice- slice architecture



Source: Spartan-3 Generation FPGA User Guide, Xilinx ug331 (v1.8)

SLICE structure

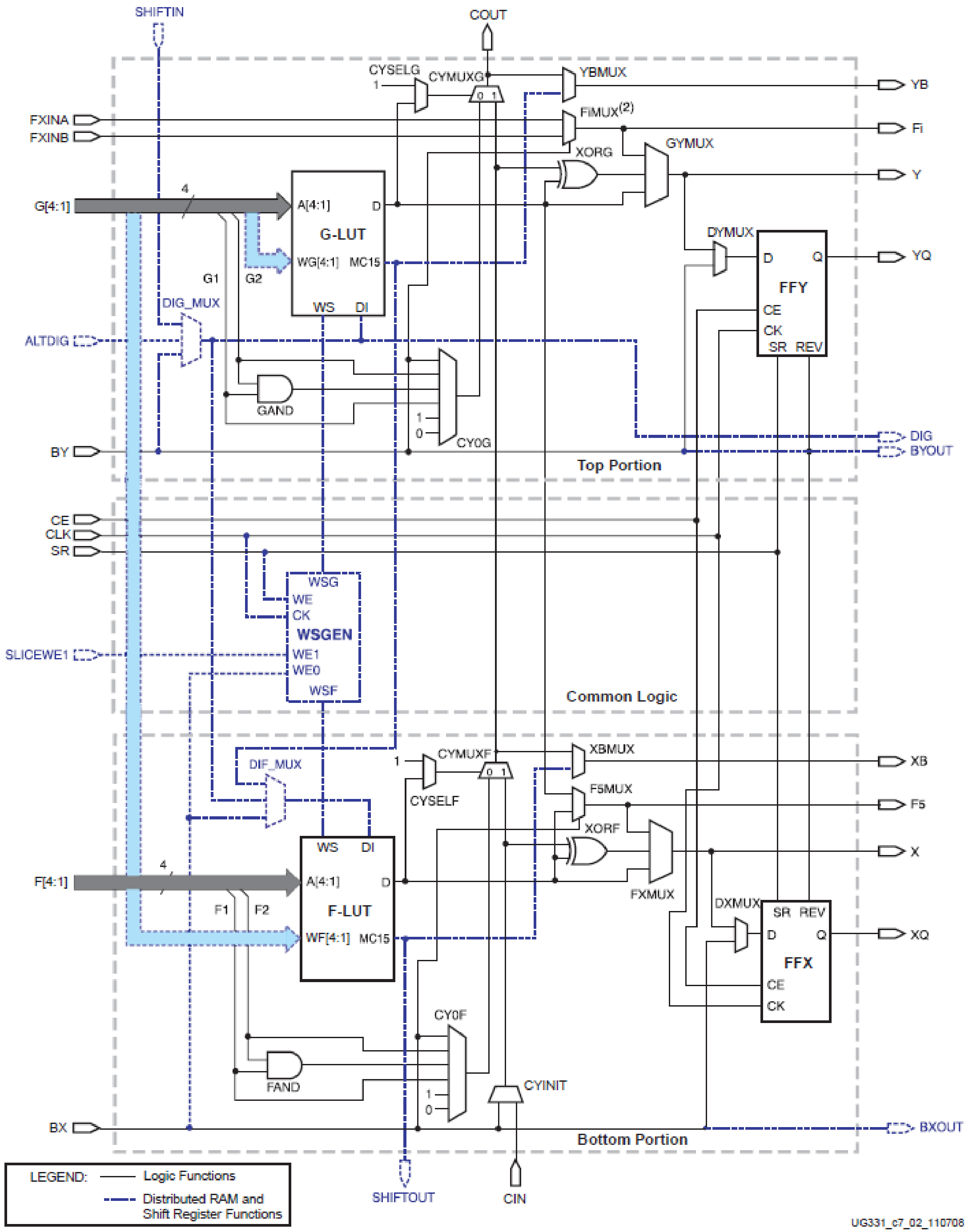
- SLICEL: two 4-input look-up tables followed by two D flip-flops plus carry & control logic (for arithmetic operations)
- SLICEM: like SLICEL, but with some extra logic to enable LUT4s to instead be used as RAM or a shift register



DS001_04_060100

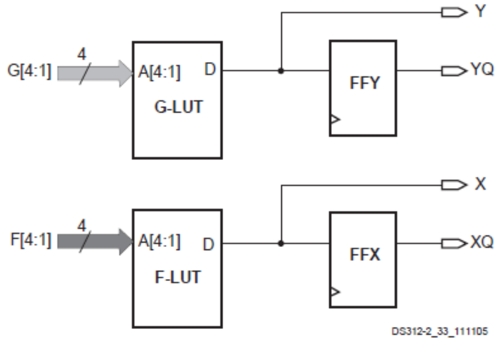
SLICE structure

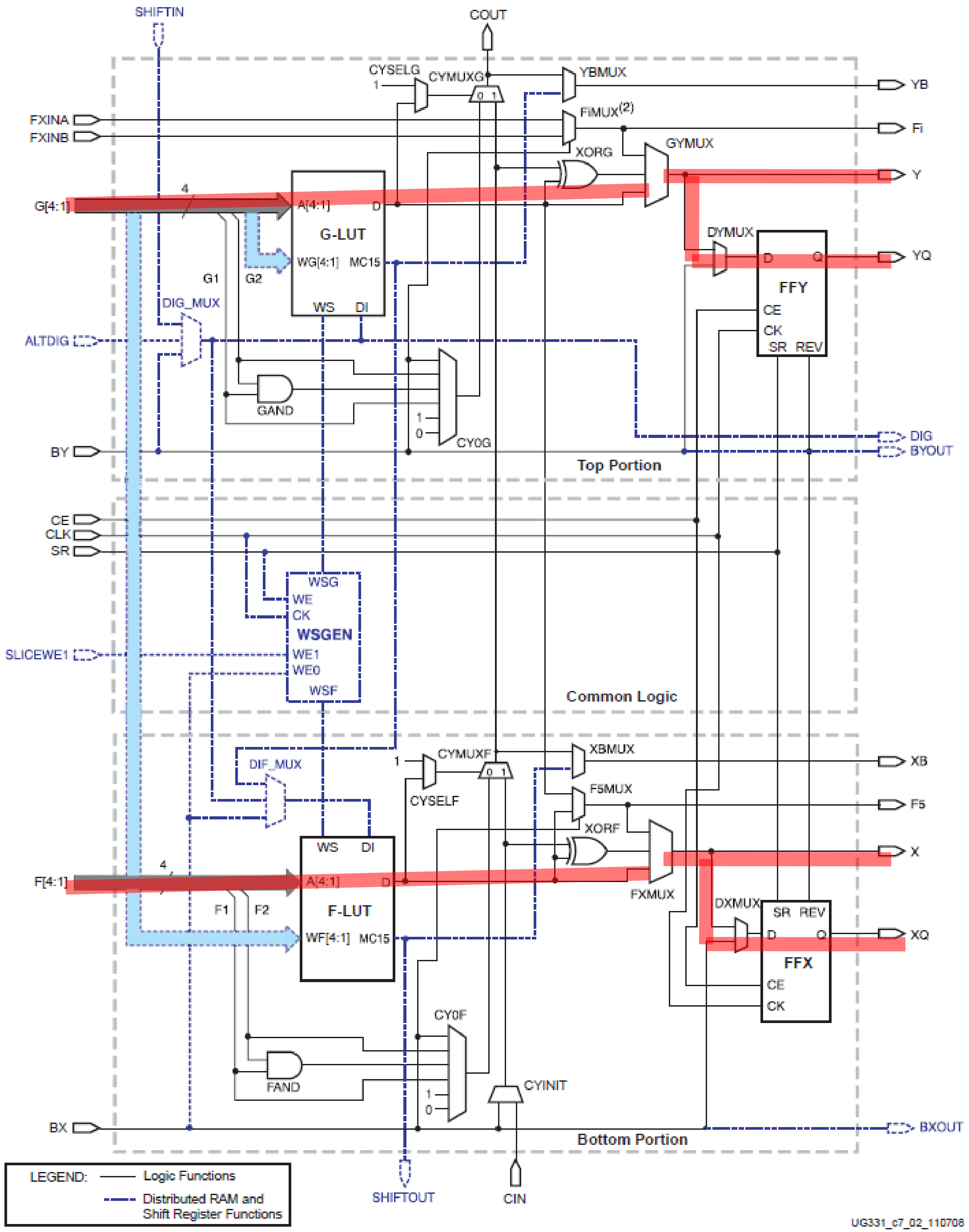
- The details of the slice architecture
- More detailed than the figure in the previous slide!
  - The control and carry logic are shown
- The blue parts are the extra logic of SLICEM



Main logic paths

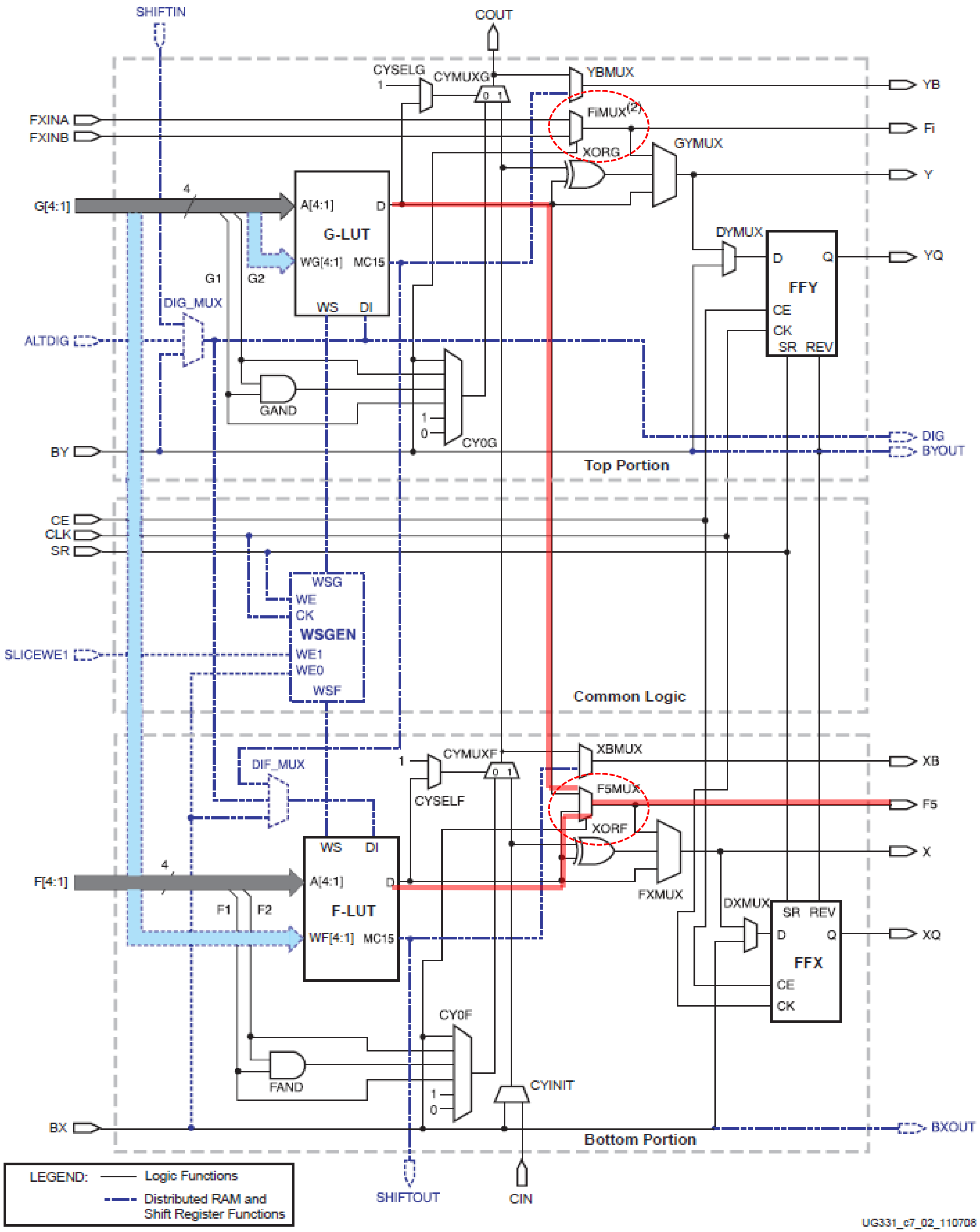
- Two 4-input functions, combinational or sequential output





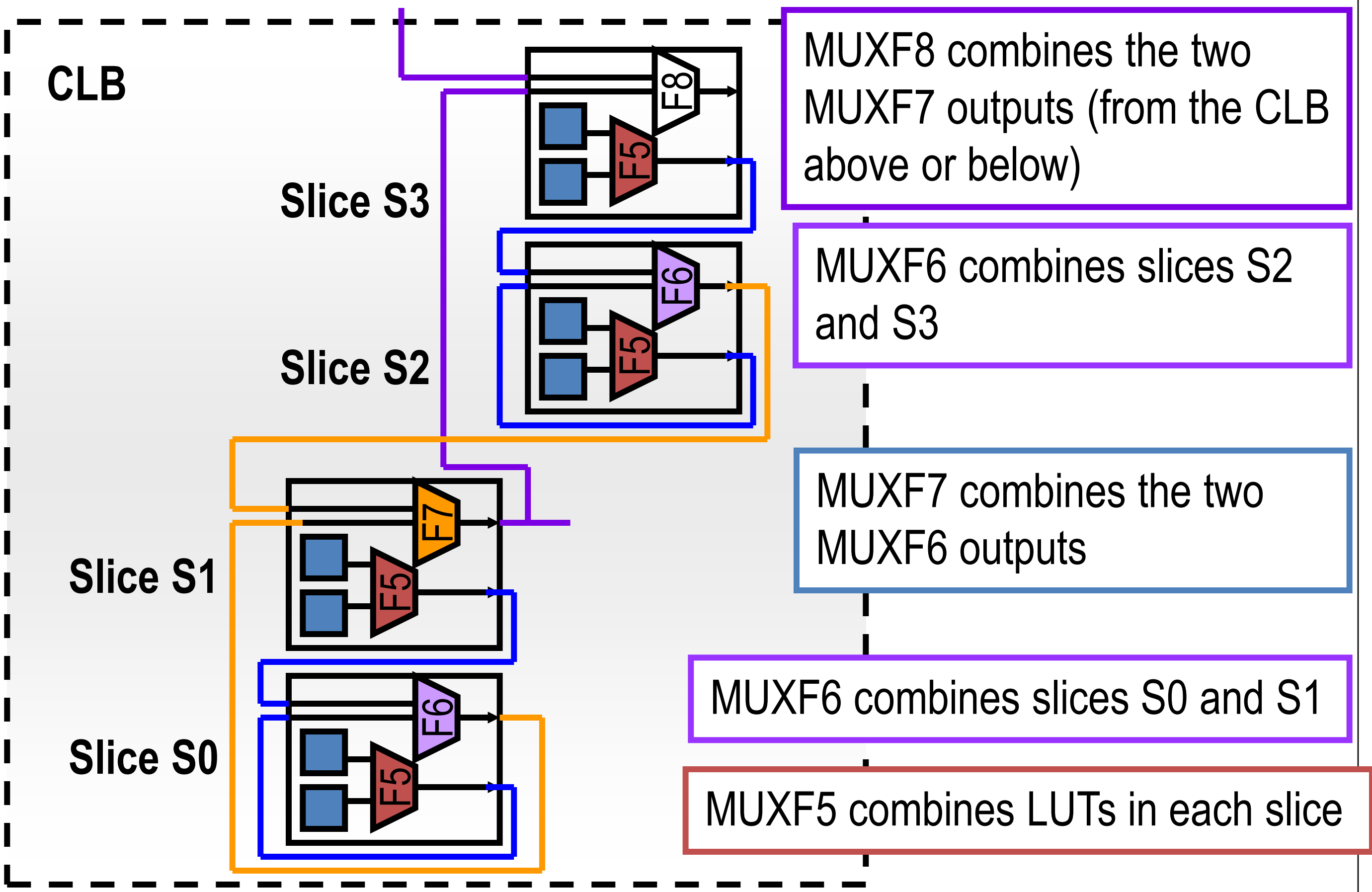
Wide multiplexers

- Combine LUTs to build more complex logic operations
- Each slice has two of these multiplexers
    - F5MUX in the bottom
    - FiMUX in the top



Wide multiplexers in a CLB

- F5MUX : multiplexes the two LUTs in a slice
- F_iMUX :
- i=6,7,8
- multiplexes the F5MUX and FiMUX of the same slice or other slices



MUXF8 combines the two MUXF7 outputs (from the CLB above or below)

MUXF6 combines slices S2 and S3

MUXF7 combines the two MUXF6 outputs

MUXF6 combines slices S0 and S1

MUXF5 combines LUTs in each slice

Cascading LUTs

- Previous slide used wide multiplexers to implement large functions
- It can implement any function with 4, 5, 6, 7, and 8 inputs
- Some functions can be re-formed and implemented by cascading LUTs, rather than combining with multiplexers
  - May use smaller number of LUTs
  - Not applicable always, depends on the function

Cascading LUTs

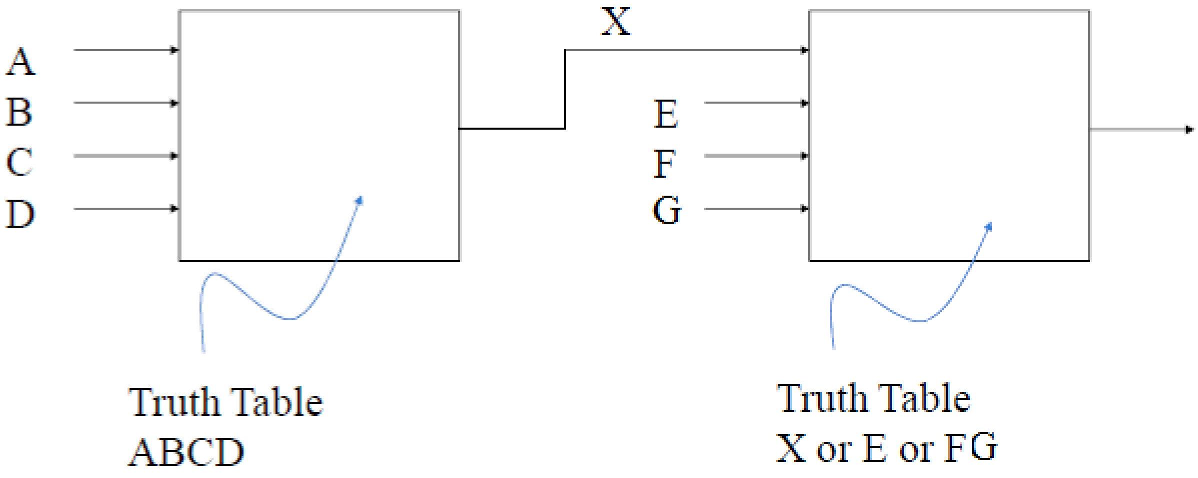
- Example: 2 cascaded LUTS for a 7-input function.
- How many LUTs when using wide MUXes?

Y = ABCDE or ABCDFG

Y = (ABCD) and (E or FG)

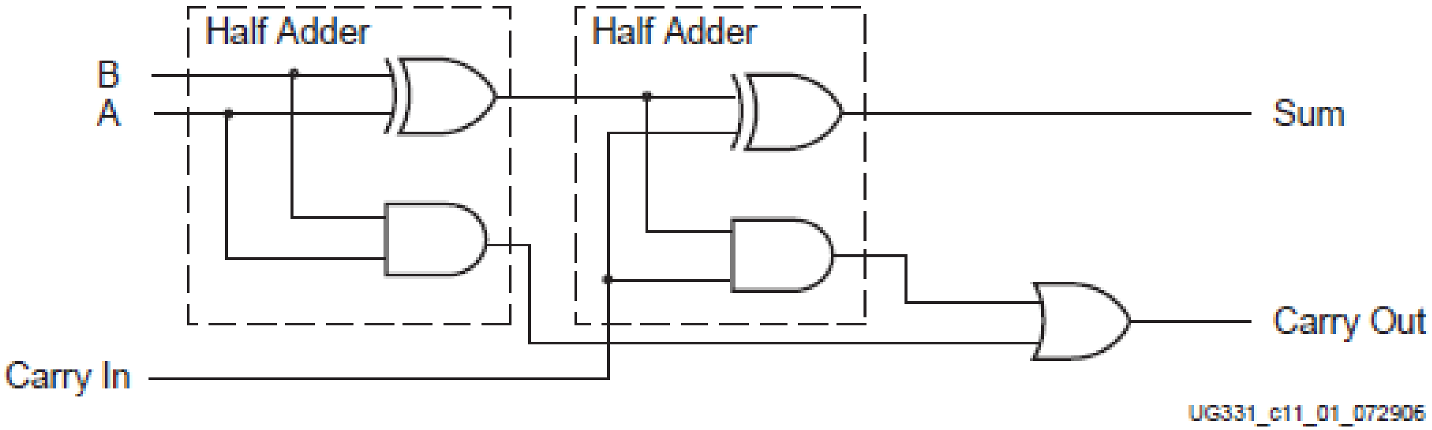
ABCD = X

Y = X and E or FG

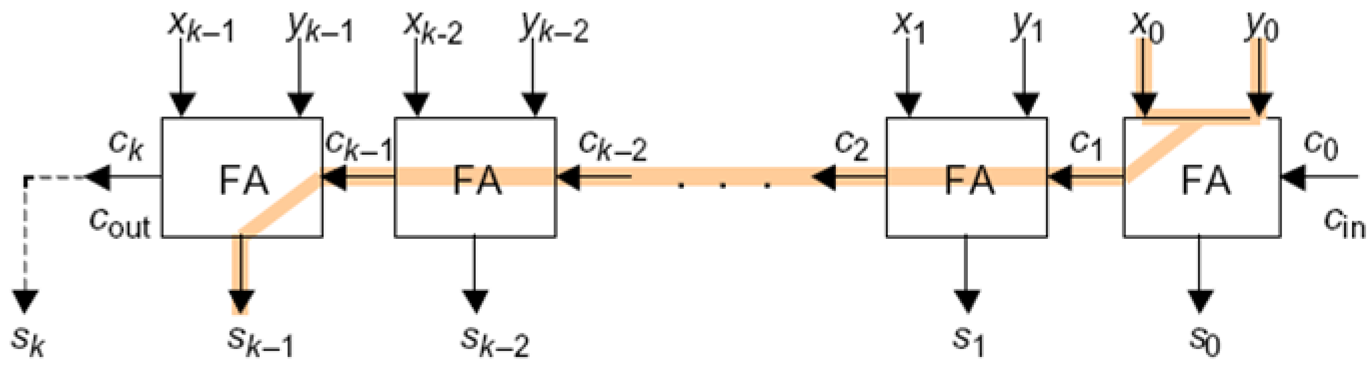


CLB arithmetic logic

- Adder is the base of many arithmetic operations
- Special logic in CLBs to implement adder
- Adders are traditionally constructed by cascading full adders
- Good modularity but long latency due to the serial operation
  - Carry must be propagated from the first to the last full adder: the $n^{th.}$ adder cannot start until the carry of the $n\text{-}1^{th.}$ full adder is calculated



Ripple carry adder

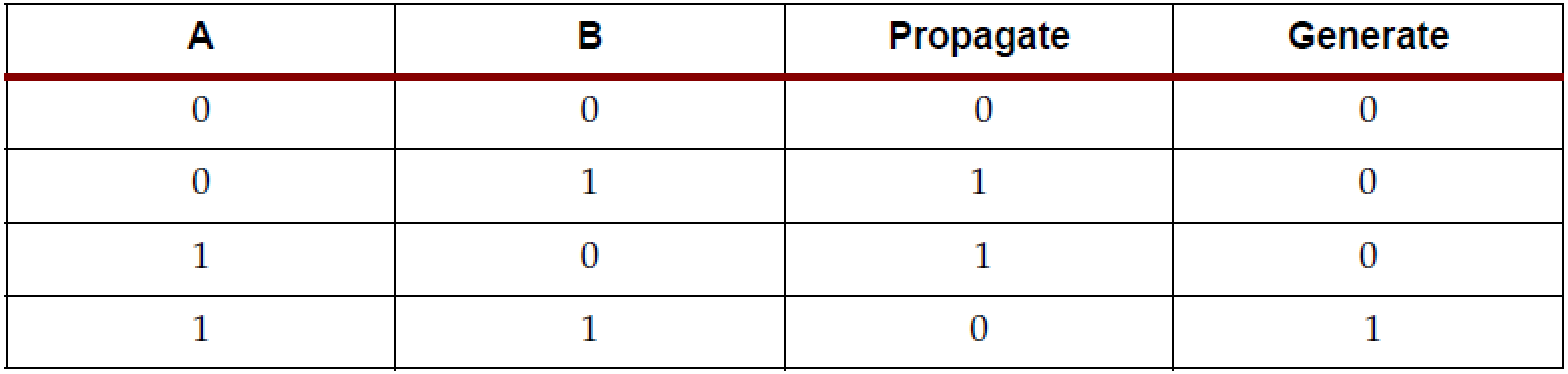


- Critical path in a k-bit ripple carry adder

Source: ECE 645, Lecture 3

CLB arithmetic logic

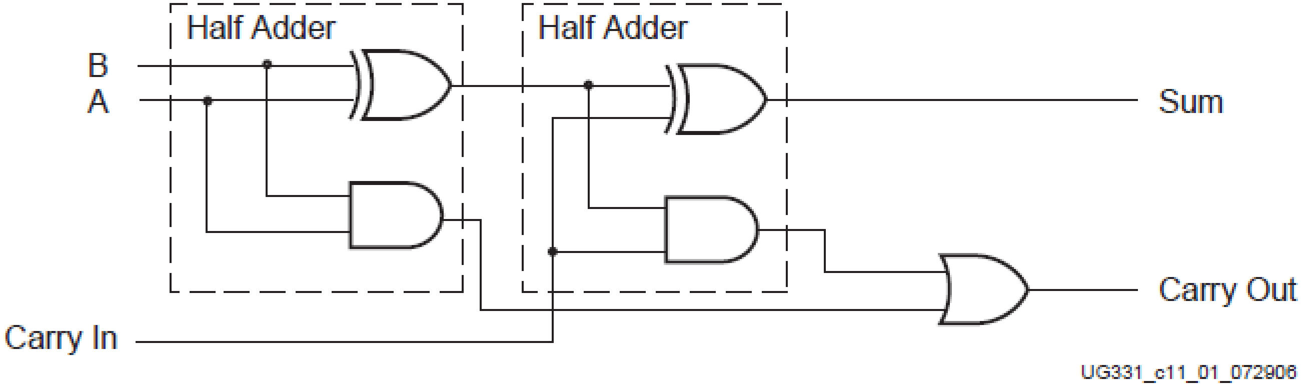
- We can predict the carry of each full adder (FA) from its inputs
- If a=b=0, no carry will be produced by the FA (regardless of the carry_in)
- If a=b=1, a carry will be produced by the FA (regardless of the carry_in)
- If a≠b, the input carry will be propagated to the output



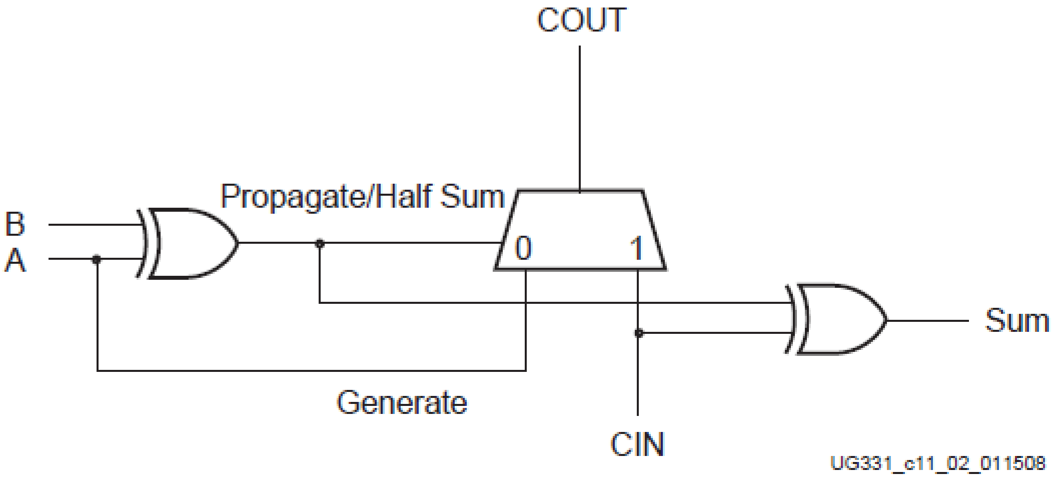
A	B	Propagate	Generate
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Carry look-ahead adder

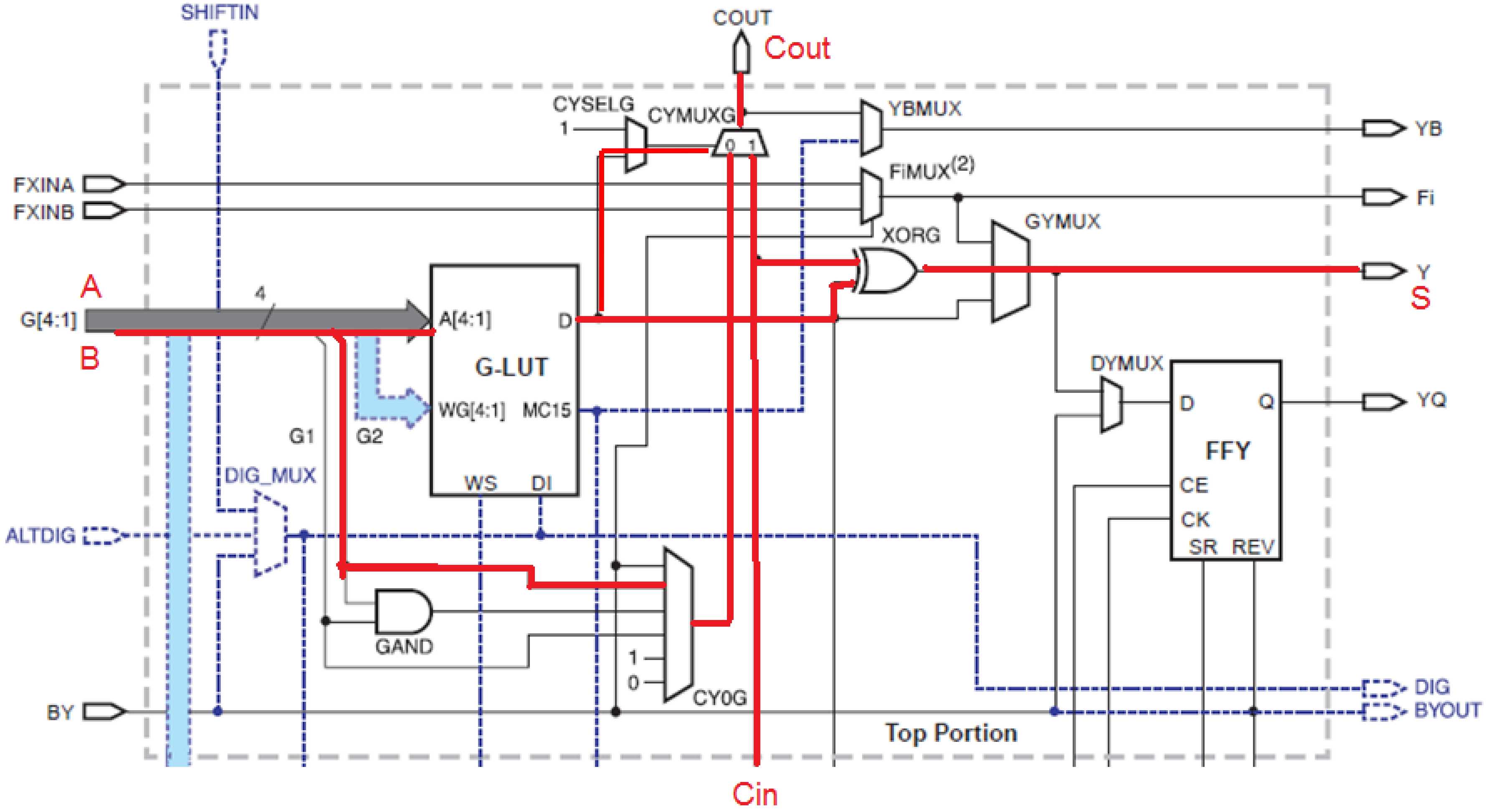
Conventional serial full adder



Full-adder with carry look-ahead Implemented in SPARTAN-3



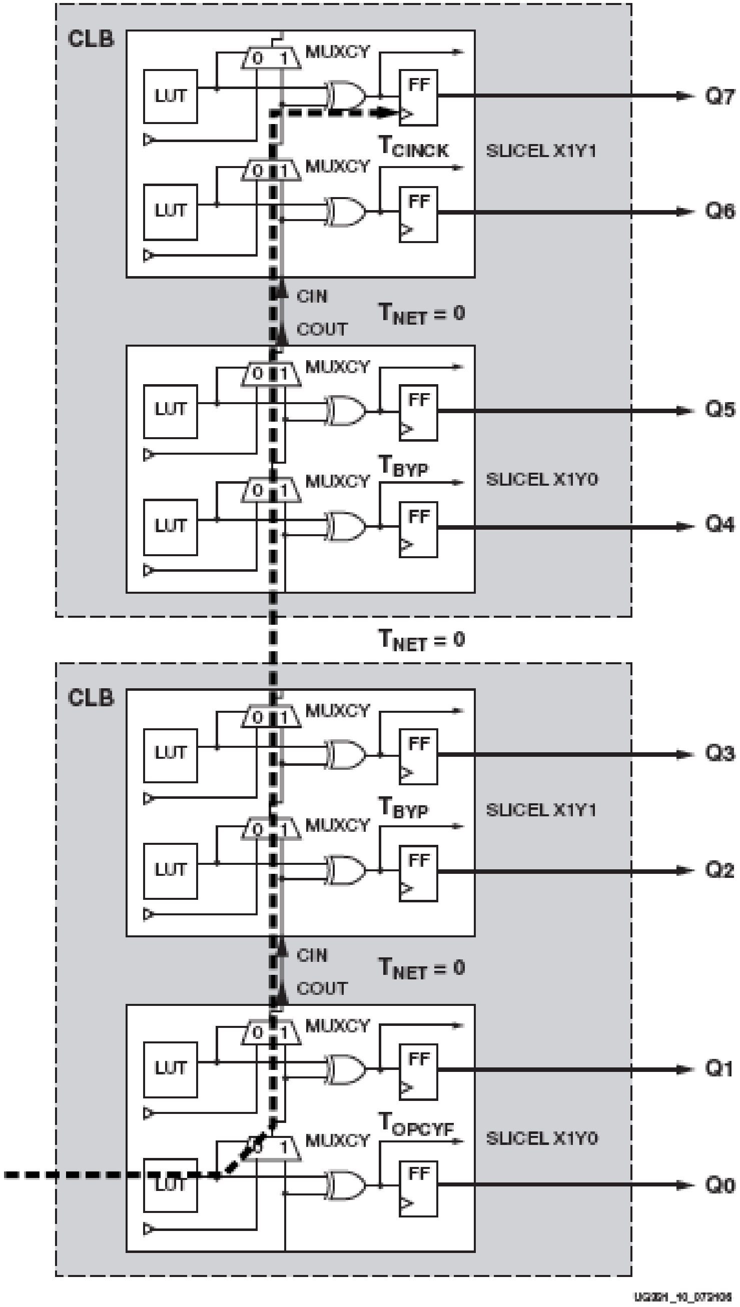
Carry look-ahead adder in SPARTAN



- Adder data-path in one slice (the upper part of slice is shown)
- Implements an FA to add two 1-bit inputs A and B

Adder by Spartan-3

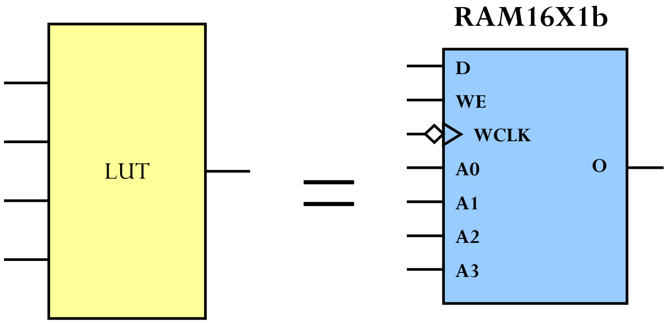
- 8-bit adder using 4 slices of Spartan-3

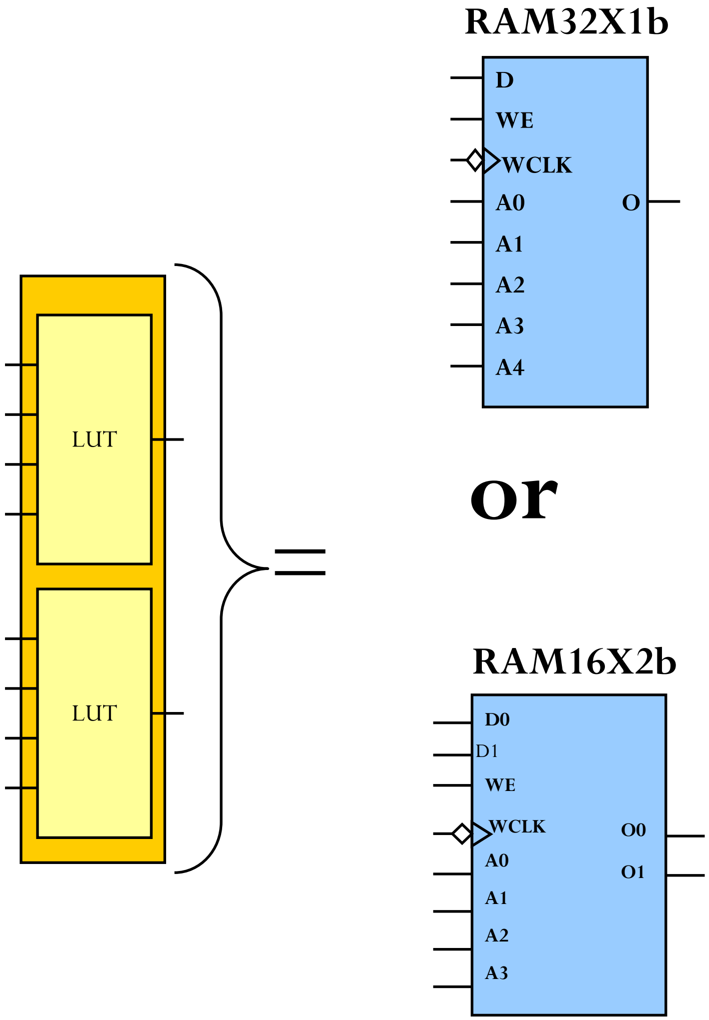


Source: ECE 645, Lecture 3

Distributed RAM

- A LUT in a SLICEM may be configured for use as a RAM
  - Implement single and dual port
  - Cascade LUTs to increase size
- Synchronous write only
- Reads may be synchronous or asynchronous.





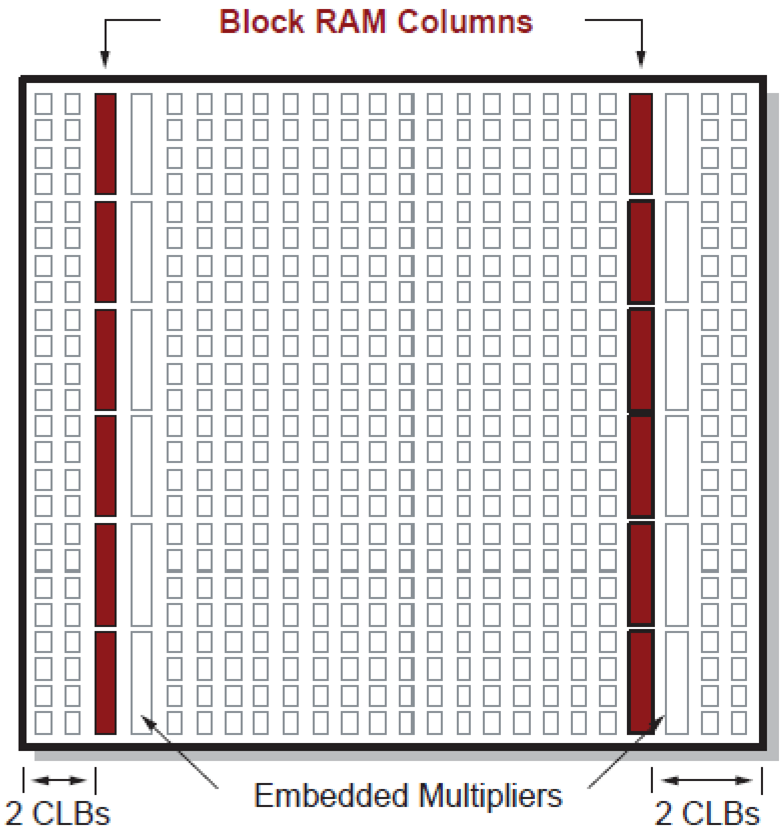
Source: EE178 Lecture 4, by T. Cauhuu

Spartan-3 components

- Configurable Logic Blocks (CLB)
- Block RAM
- Programmable Input Output Blocks (IOB)
- Dedicated multipliers

Block RAM

- On-chip memory blocks of Static RAM
- Both single and dual-port memories
- XC3S400 has two columns of RAM
- Each column contains 8 blocks of RAM (the figure shows 6 blocks!)
- Each block is 18-kilobit synchronous RAM
  - Each block has 16 kb data, 2kb parity
  - Total RAM=18 kb×16 blocks= 288 kb



Block RAM organization

- Can be used as RAM blocks with different lengths and widths
- 2kb of the RAM is devoted to parity bits
- 16Kx1 configuration can be used as a large lookup table

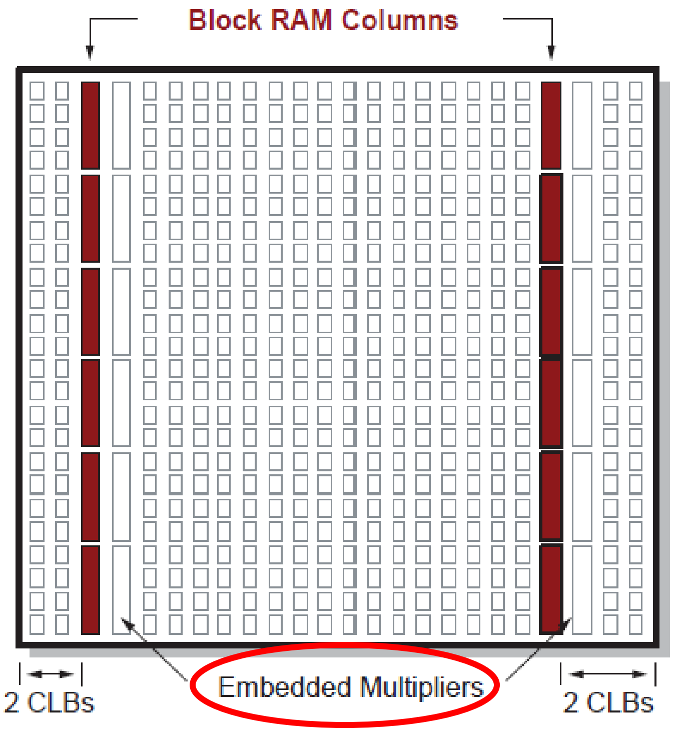
Configuration	Depth	Data bits	Parity bits
16K x 1	16Kb	1	0
8K x 2	8Kb	2	0
4K x 4	4Kb	4	0
2K x 9	2Kb	8	1
1K x 18	1Kb	16	2
512 x 36	512	32	4

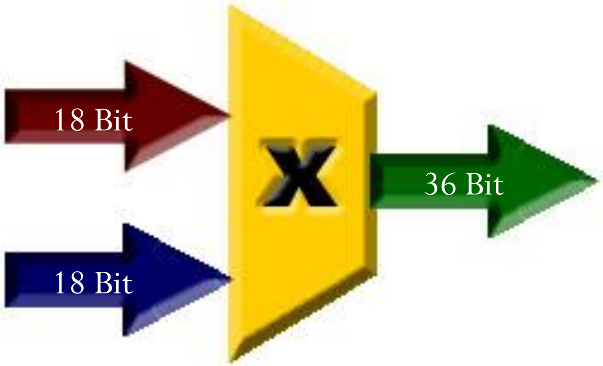
Spartan-3 components

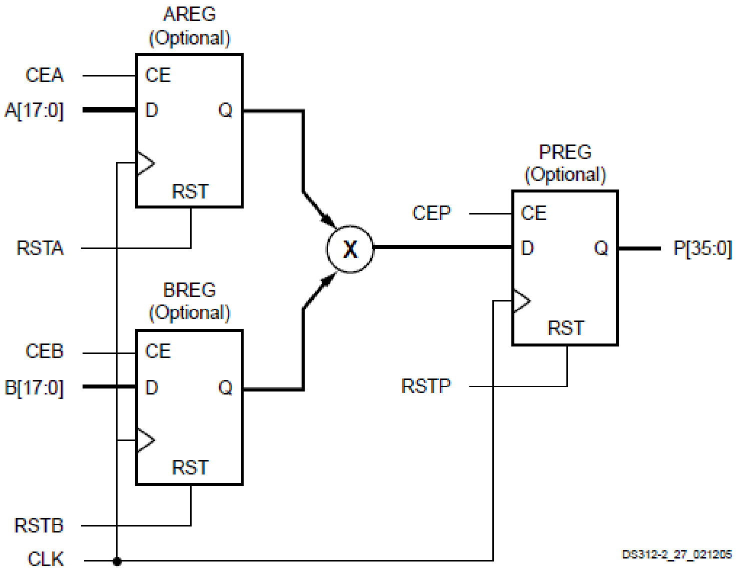
- Configurable Logic Blocks (CLB)
- Block RAM
- Dedicated multipliers
- Programmable Input Output Blocks (IOB)

Multiplier

- 18×18 multipliers
- Very fast: single cycle latency
- 16 multipliers in Spartan-3
- Can use registers in input and output of each multiplier
  - For pipelined operation





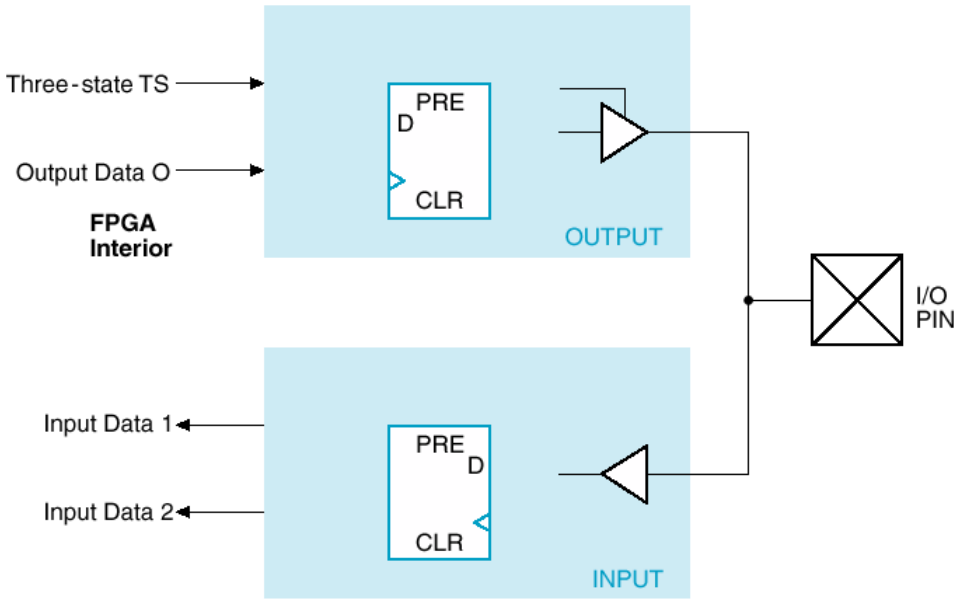


Spartan-3 components

- Configurable Logic Blocks (CLB)
- Block RAM
- Dedicated multipliers
- Programmable Input Output Blocks (IOB)

I/O resources

- 264 I/O pins in Spartan-3
- Can be configured as both input and output
- I/O pins have different architectural and electrical aspects
  - The most interesting aspect is the double data rate capability

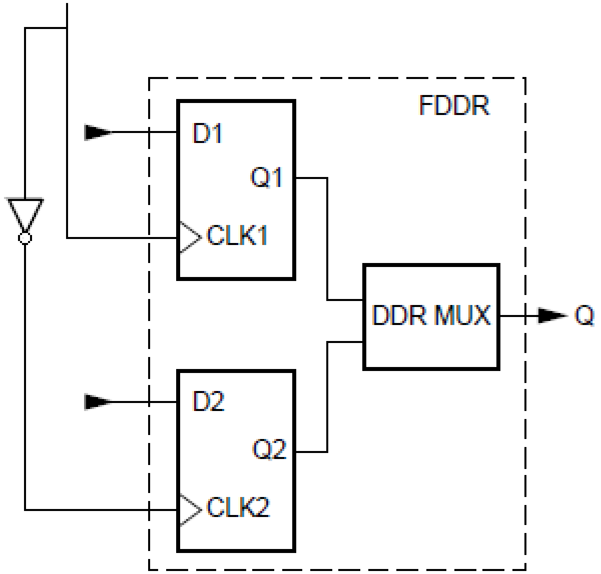


Basic I/O block

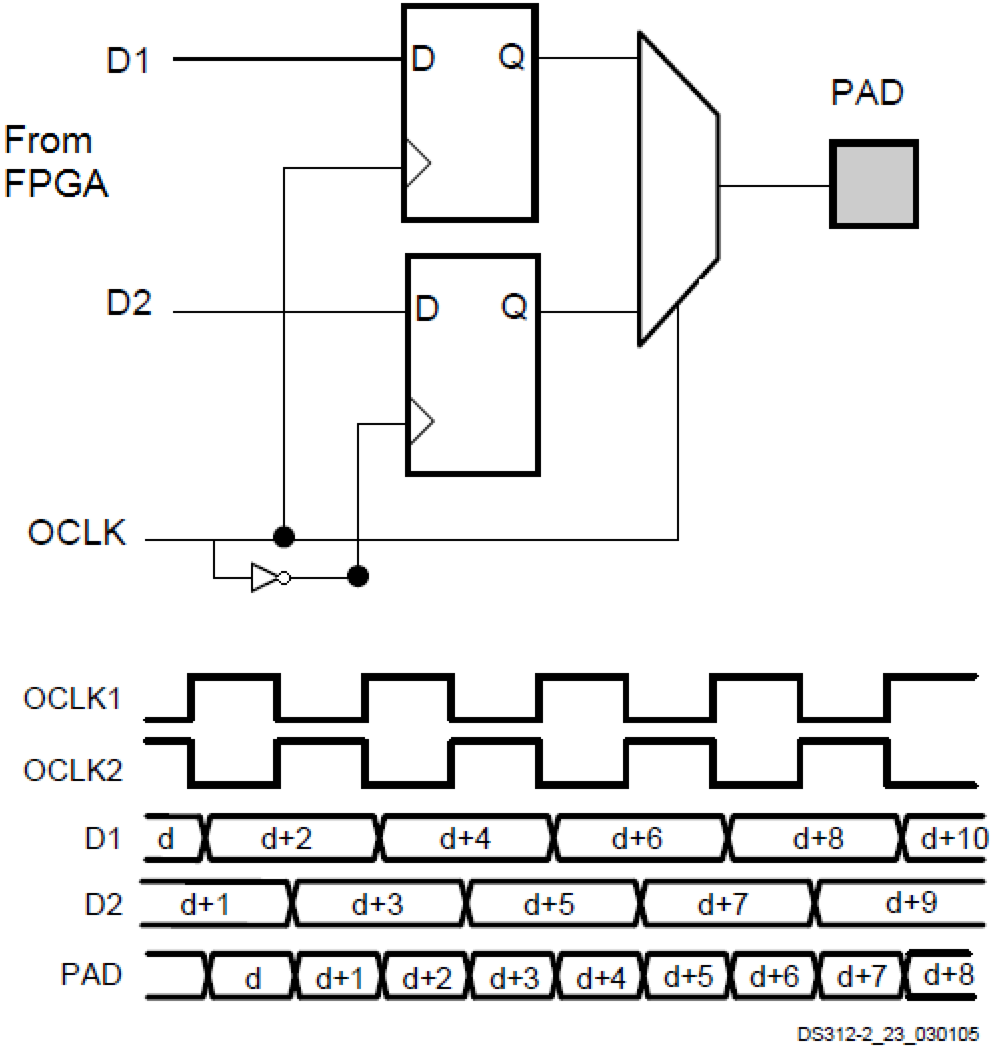
More complex blocks in reality

Spartan-3 I/O- DDR

- Double-Data-Rate (DDR) transmission describes the technique of sending data on both the rising and falling edges of the clock
- Doubles the bandwidth



Spartan-3 I/O- DDR



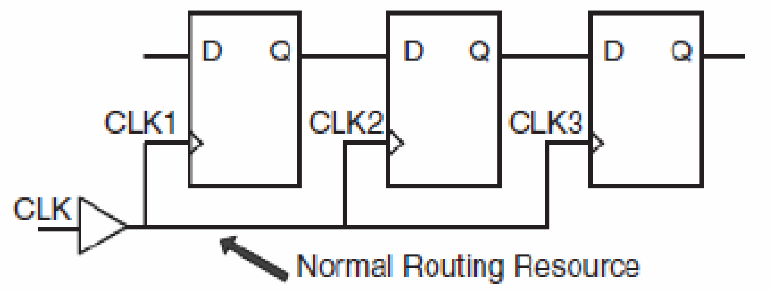
Spartan-3 components

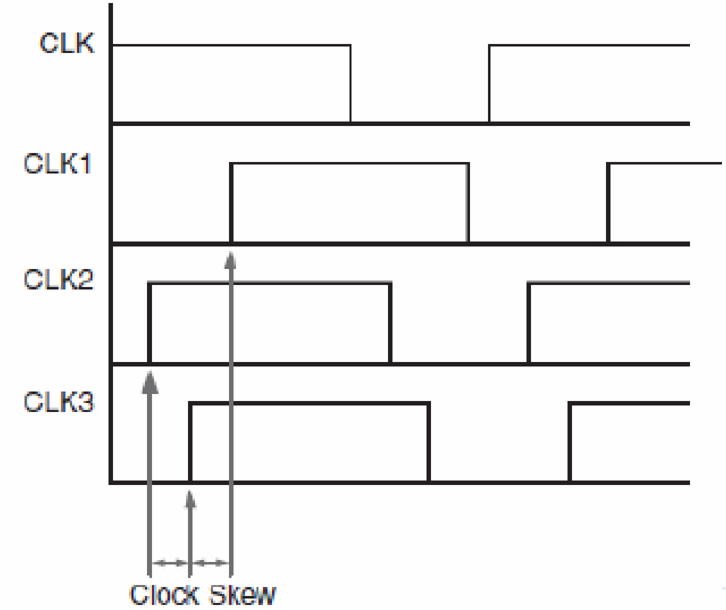
- Configurable Logic Blocks (CLB)
- Block RAM
- Dedicated multipliers
- Programmable Input Output Blocks (IOB)
- Digital Clock Manager (DCM)

Clocking in FPGAs

- Clock skew is an important problem in clock distribution

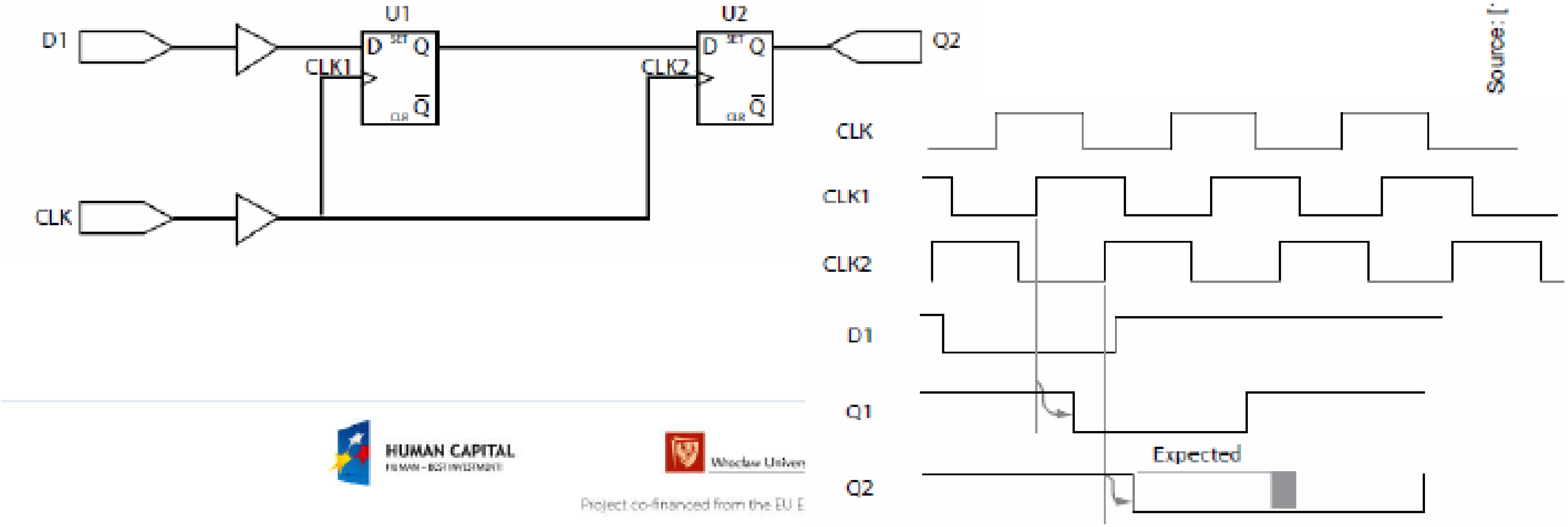
  - The clock ***skew***, S, is the maximum delay from the clock input of one flip-flop to the clock input of another flip-flop.





Clocking in FPGAs- skew problem

- Clock skew is an important problem

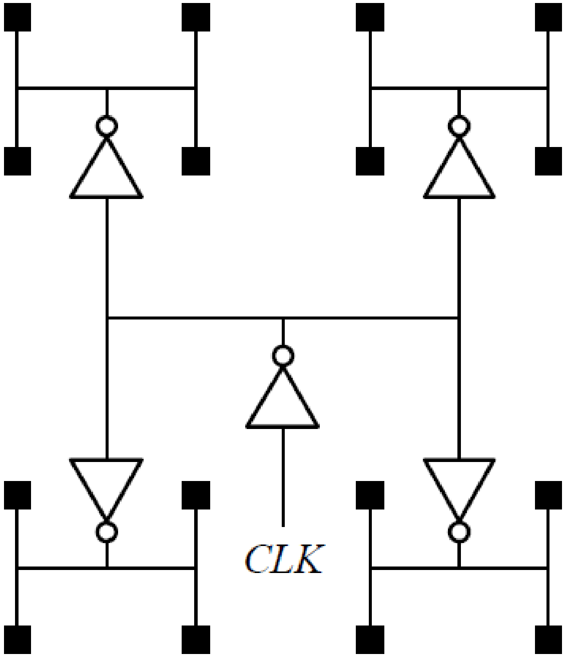


Source: [

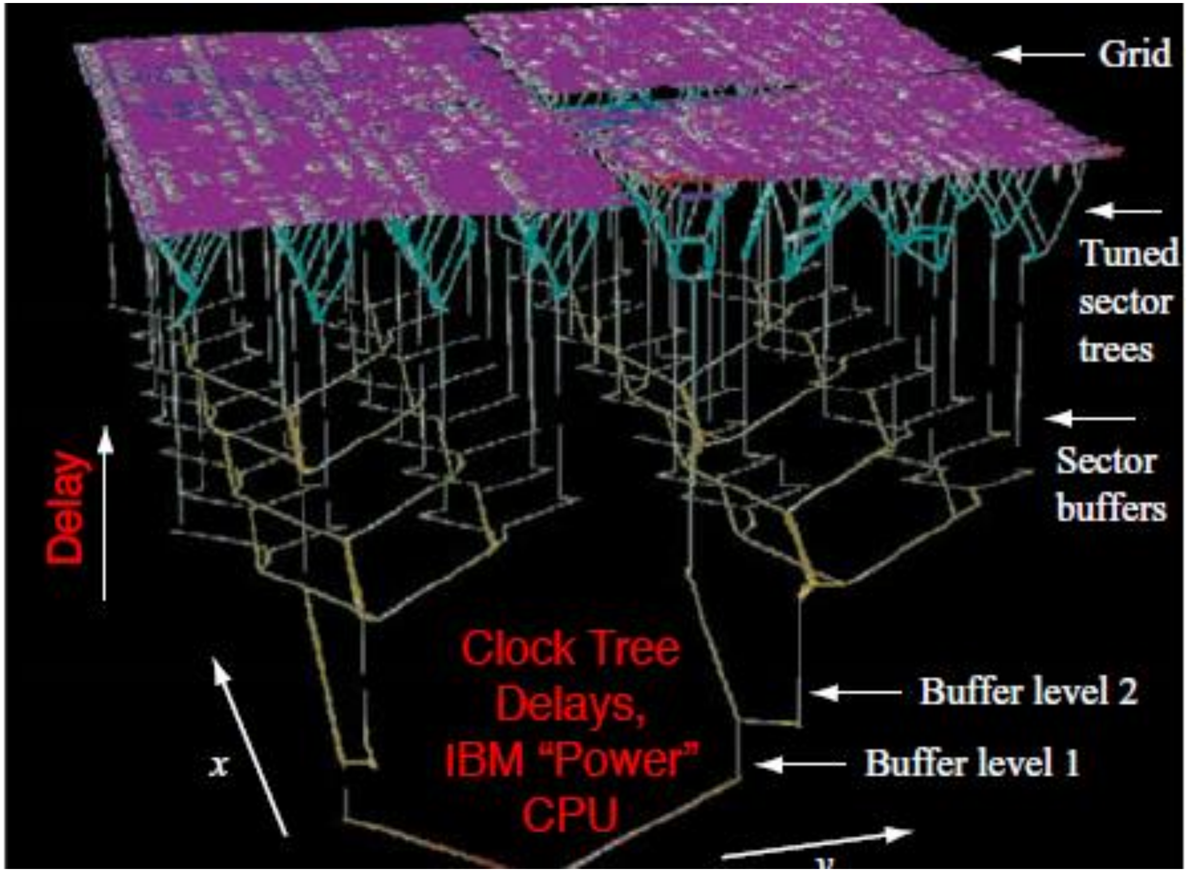
Skew solution

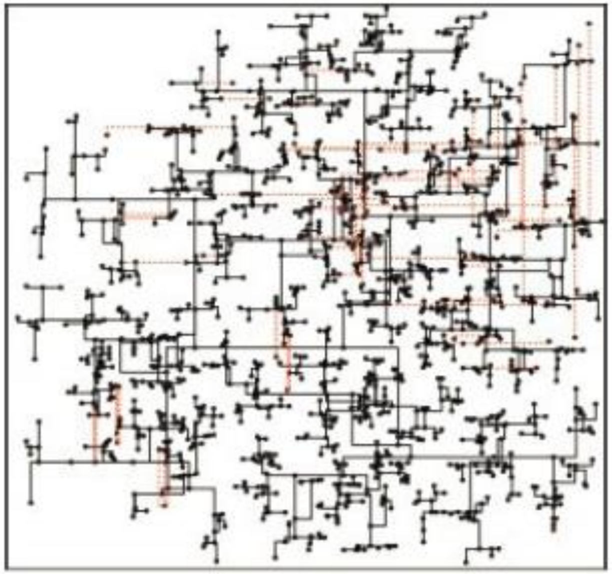
- Don’t use gated clocks
- Balance clock paths (tree distribution)
- The most common way: H-tree:
  - All FFs (black squares)

    receive clock at the same time

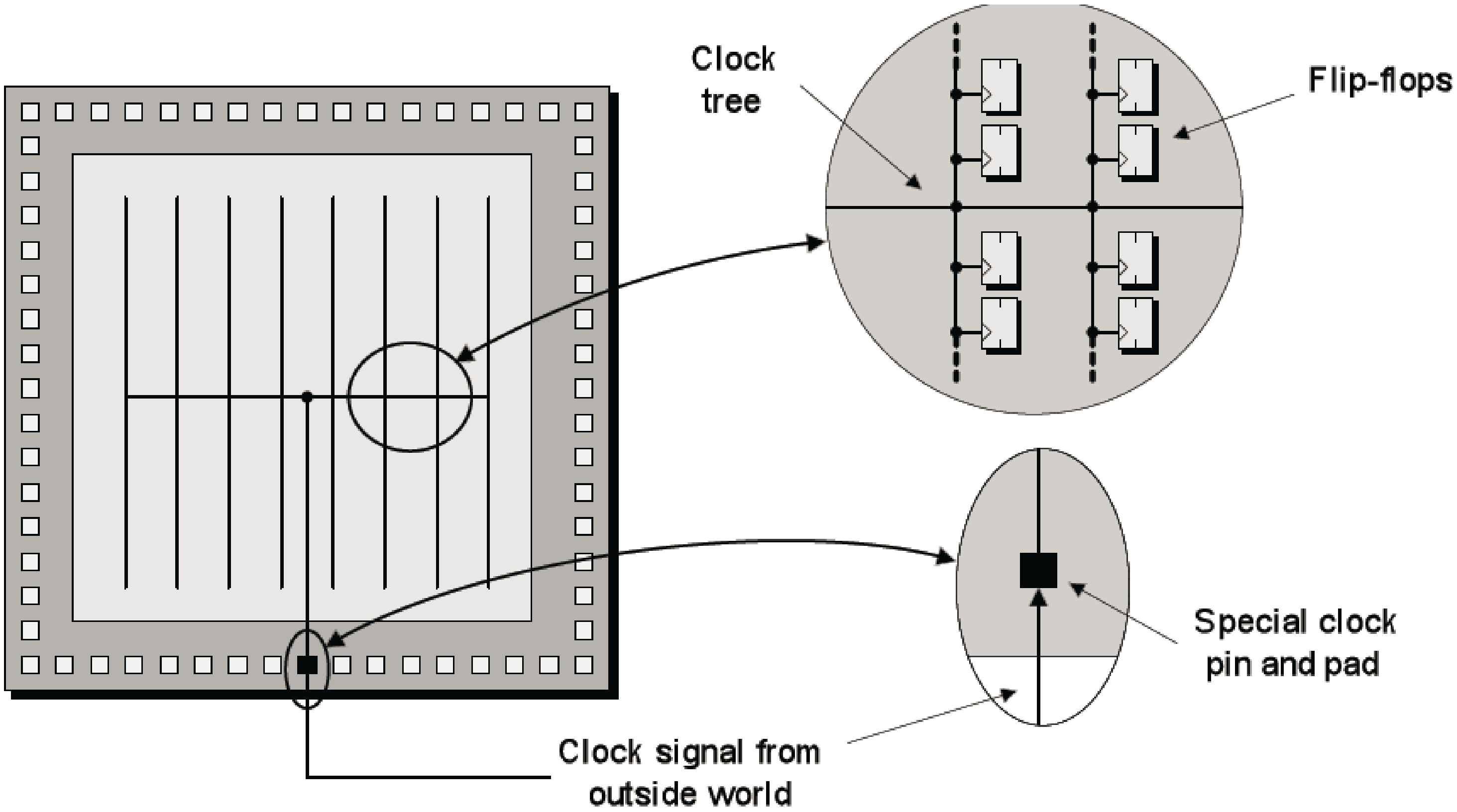


Realistic H-trees



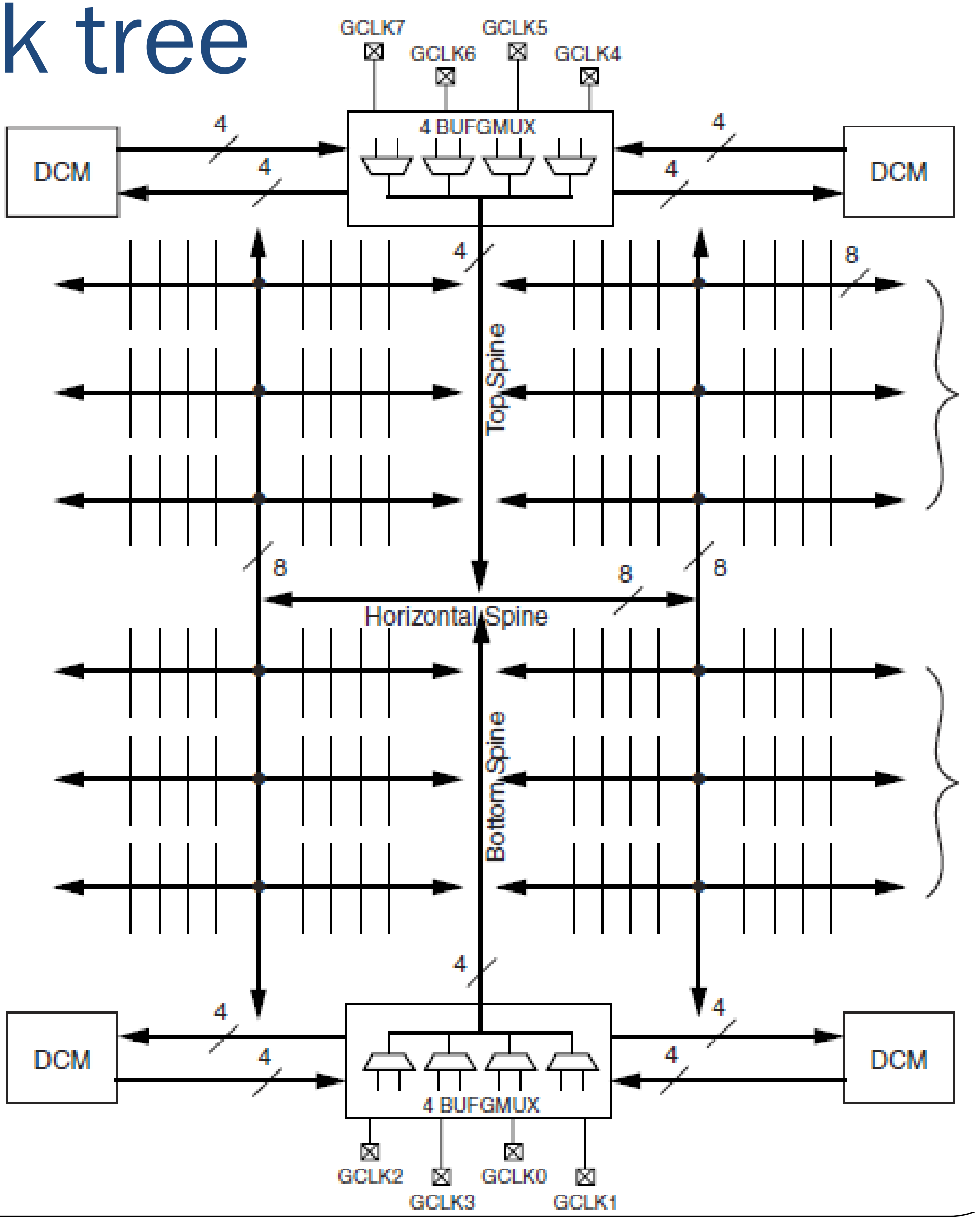


FPGA clock tree



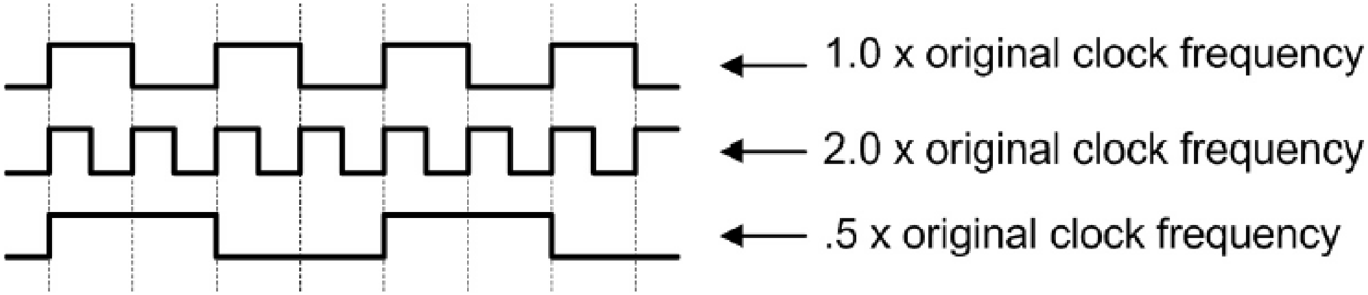
Spartan-3 clock tree

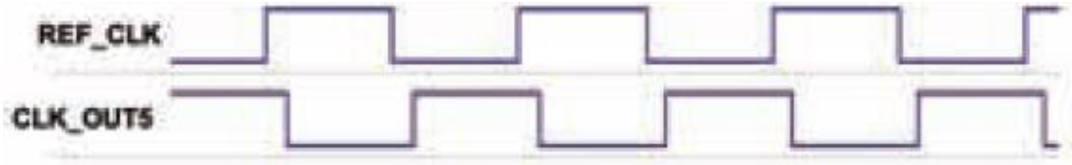
- Spartan-3 devices have eight Global Clock input pads called GCLK0 through GCLK7
- The eight BUFGMUX components (4 at the top and 4 at the bottom side) can multiplex between clock sources
  - From input pins or from DCMs



Digital Clock Manager (DCM)

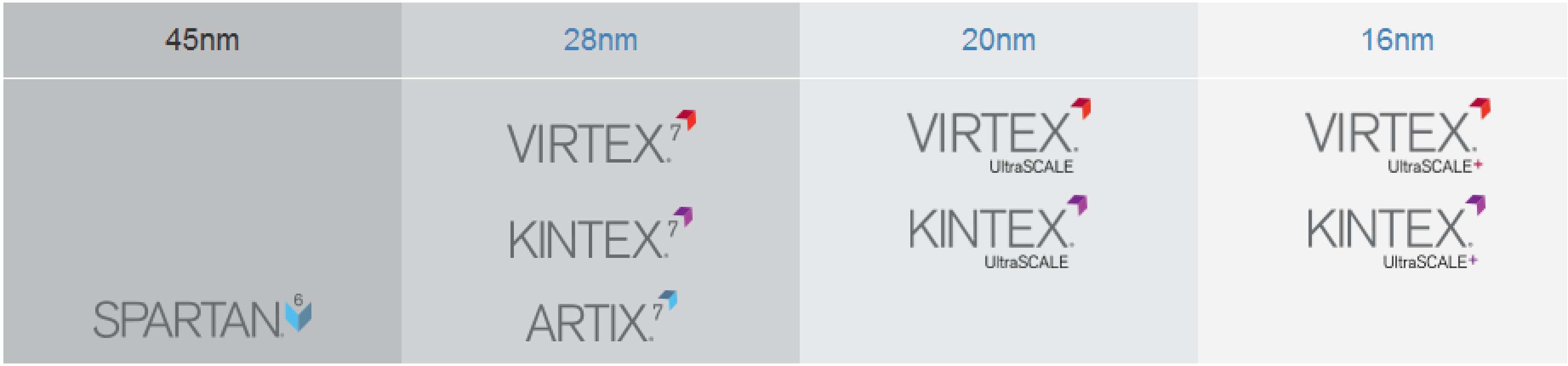
- Used to generate clock signal for FPGA out of the external clock signal
- Phase Shift a clock signal
- Multiply or divide an incoming clock frequency or synthesize a completely new frequency by a mixture of clock multiplication and division


1.0 x original clock frequency
2.0 x original clock frequency
.5 x original clock frequency



Phase shift

Modern Xilinx FPGAs



45nm	28nm	20nm	16nm
	VIRTEX 7	VIRTEX UltraSCALE	VIRTEX UltraSCALE+
	KINTEX 7	KINTEX UltraSCALE	KINTEX UltraSCALE+
SPARTAN 6	ARTIX 7		

- Virtex: high capacity (up to 2M logic cells), high-performance
- Kintex: high performance, but smaller than Virtex
- Artix: mid-range, low-cost and low-power
  - Replacement of Spartan

Modern Xilinx FPGAs

- All 7 series families share the same basic building blocks
- The mixture and number of these resources varies across families

⬜ **Logic Fabric**
LUT-6 CLB

🟥 **Precise, Low Jitter Clocking**
MMCMs

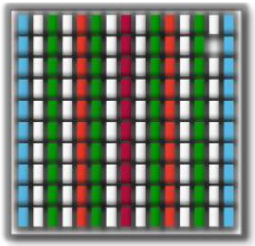
🟩 **On-Chip Memory**
36Kbit/18Kbit Block RAM

🟦 **Enhanced Connectivity**
PCIe® Interface Blocks

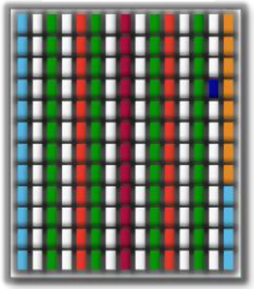
🟥 **DSP Engines**
DSP48E1 Slices

🟦 **Hi-perf. Parallel I/O Connectivity**
SelectIO™ Technology

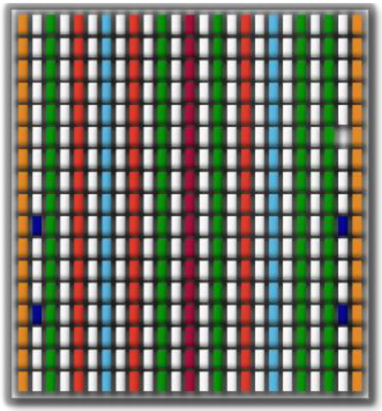
🟦 **Hi-performance Serial I//O Connectivity**
Transceiver Technology



Artix™-7 FPGA



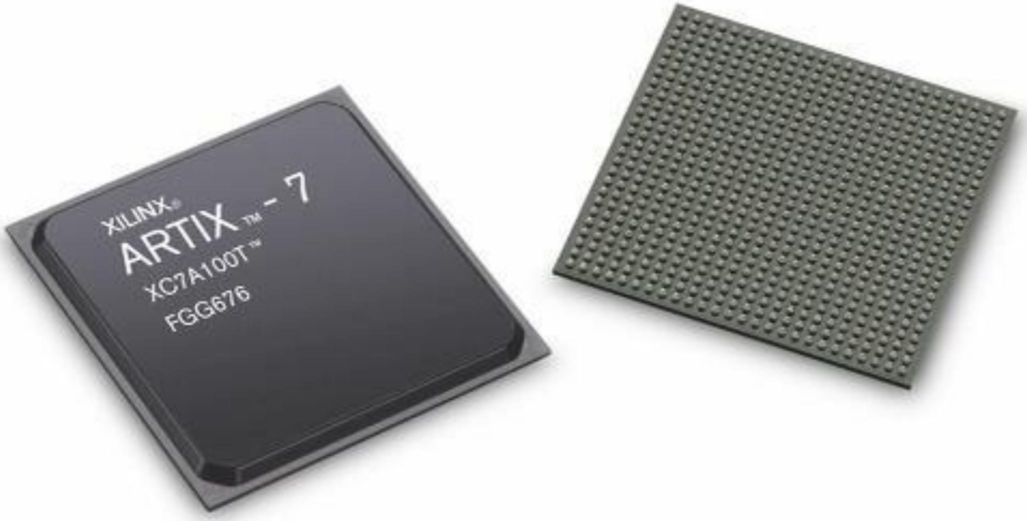
Kintex™-7 FPGA



Virtex®-7 FPGA

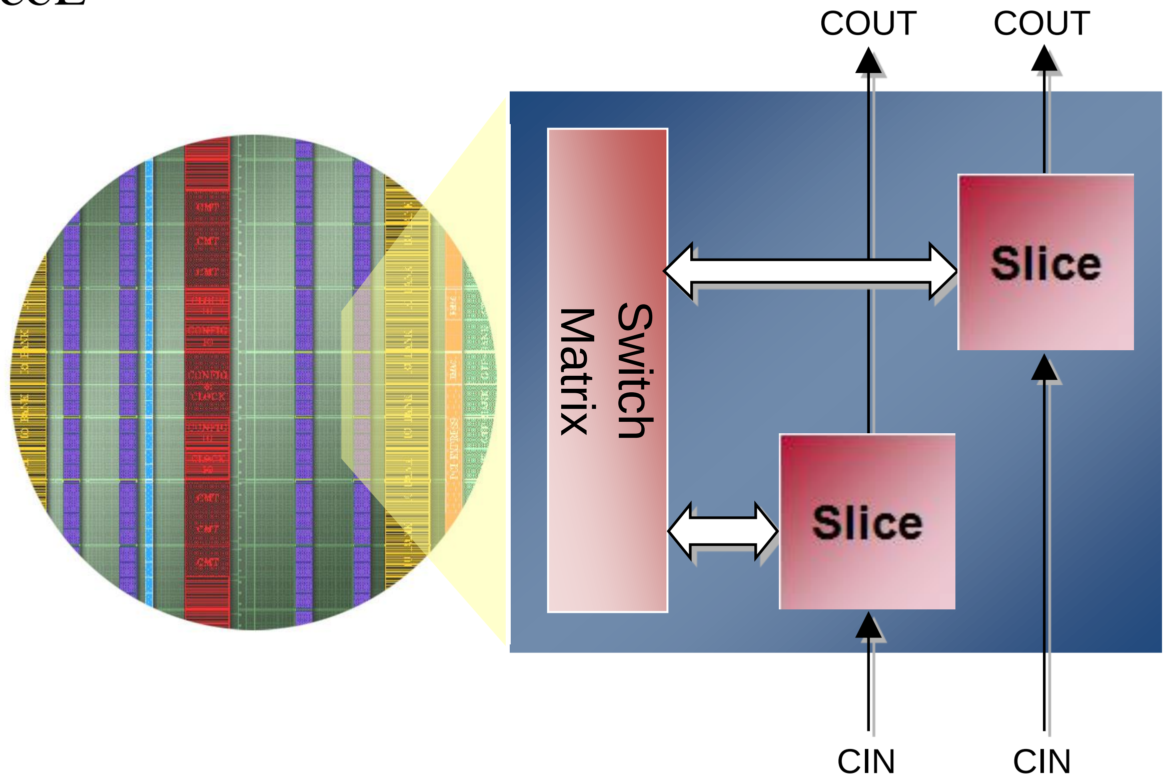
Artix-7

- Spartan-6 is the last member of the family
- Replaced by the Artix family



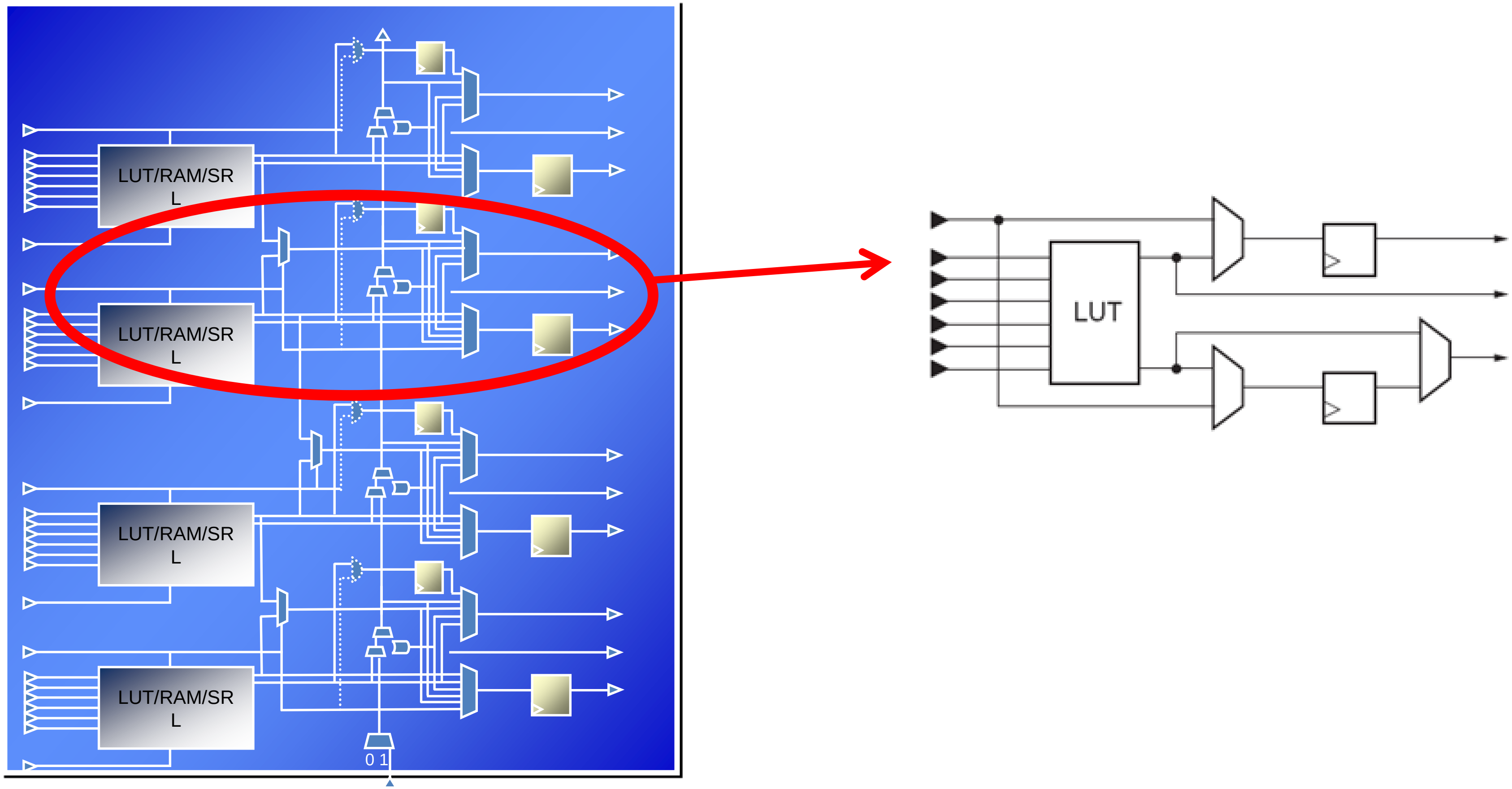
CLB in Artix-7

- Each CLB has two slices
- Slices still can be either SliceM and SliceL



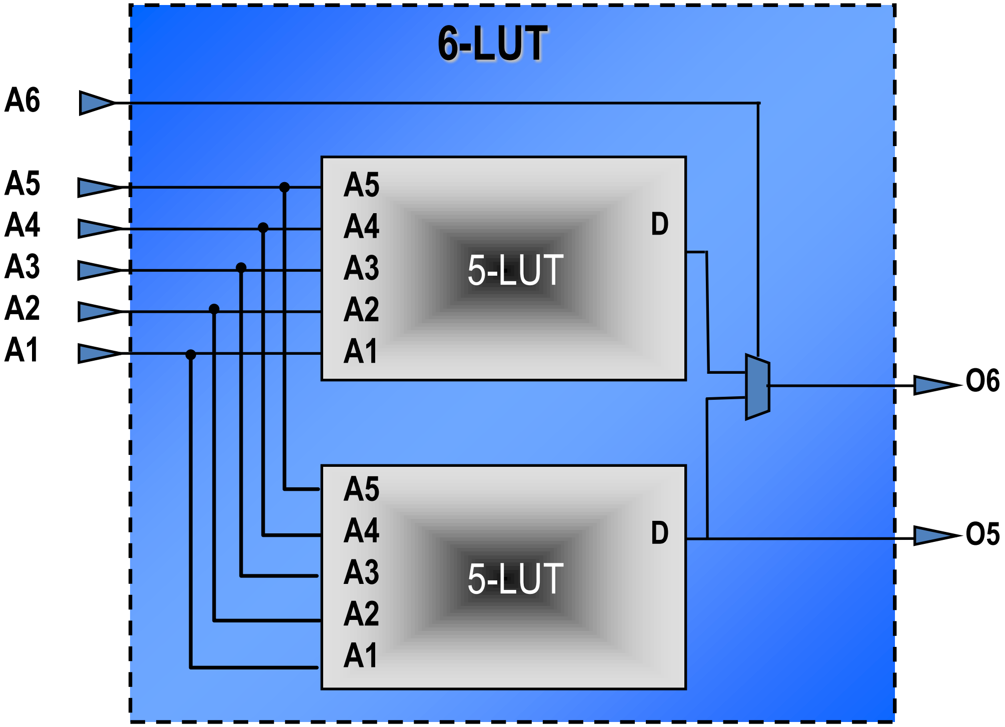
Slice structure

- Each slice has four 6-input LUTs



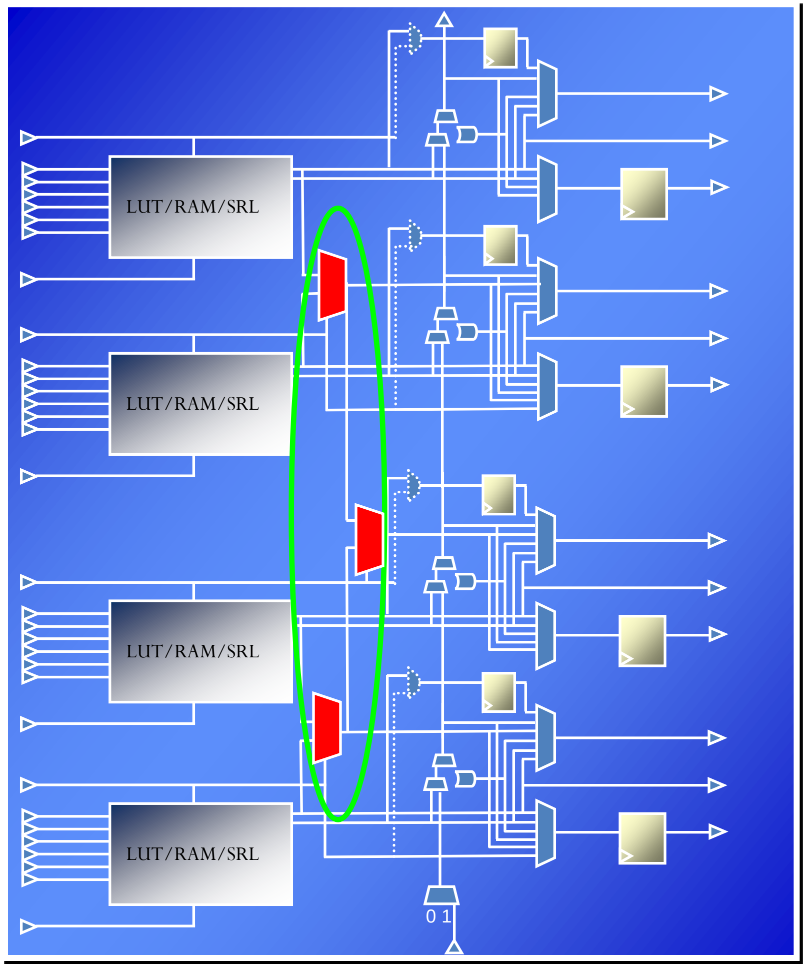
6-Input LUT with Dual Output

- 6-input LUT can be two 5-input LUTs with common inputs
  - Any function of six variables or two independent functions of five variables



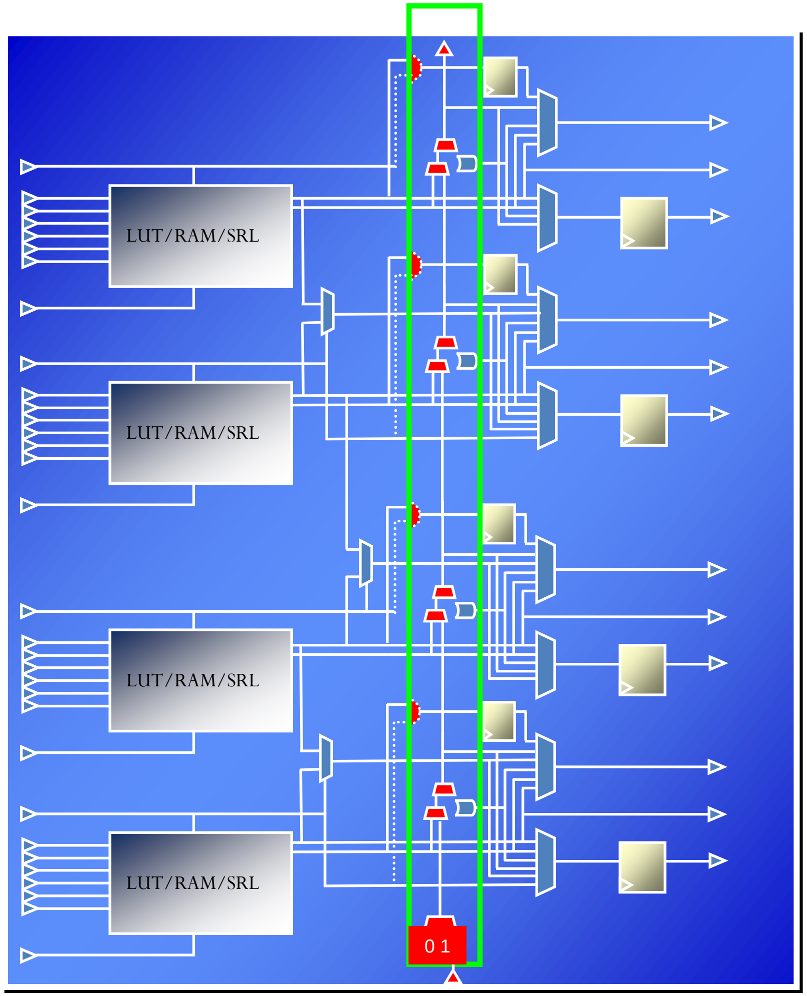
Wide Multiplexers

- Each F7MUX combines the outputs of two LUTs together
  - Can implement an arbitrary 7-input function
  - Can implement an 8-1 multiplexer
- The F8MUX combines the outputs of the two F7MUXes
  - Can implement an arbitrary 8-input function



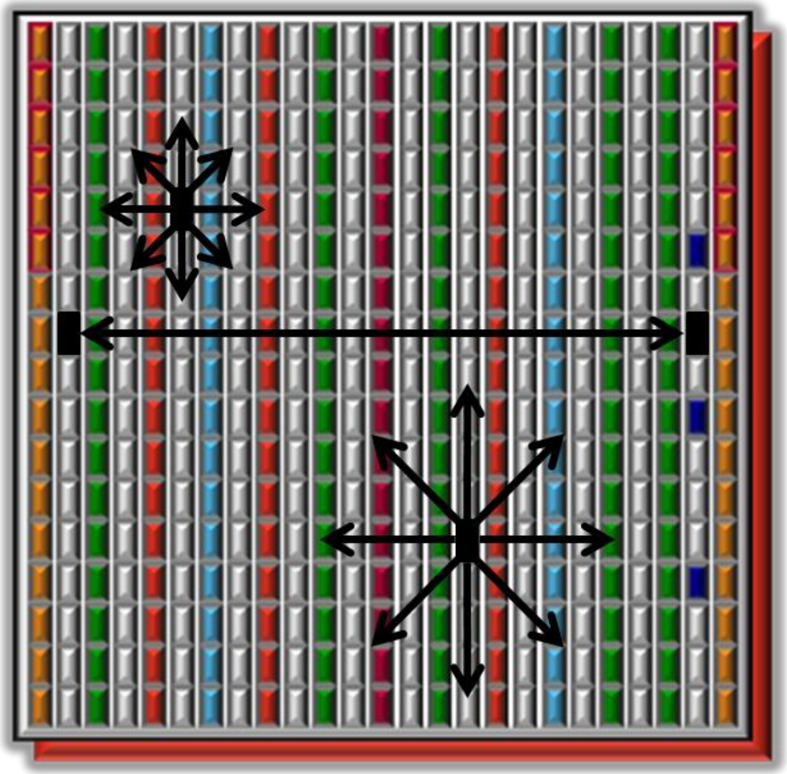
Carry Chain

- Carry chain can implement fast arithmetic addition and subtraction
  - Carry out is propagated vertically through the four LUTs in a slice
  - The carry chain propagates from one slice to the slice in the same column in the CLB above
- Carry look-ahead
  - Combinatorial carry look-ahead over the four LUTs in a slice
  - Implements faster carry cascading from slice to slice

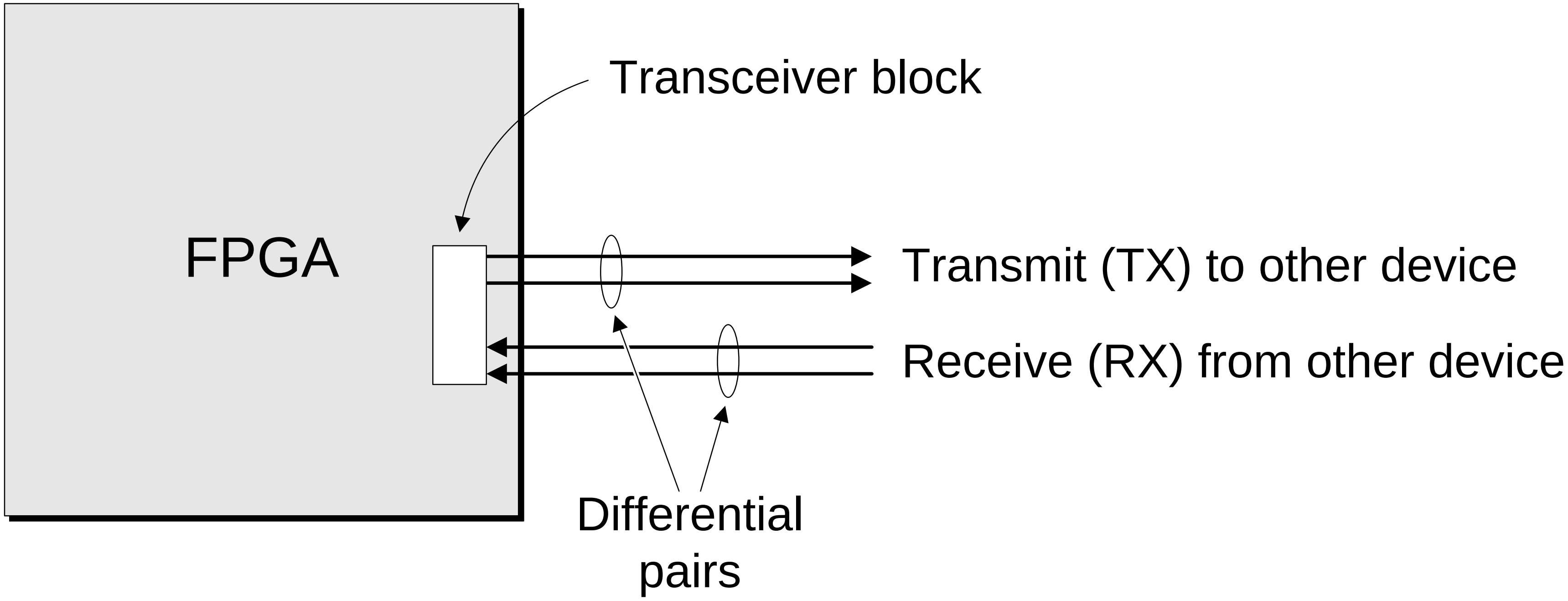


Interconnects

- Connections between CLBs and other resources use the fabric routing resources
- Routes connect resources vertically, horizontally, and diagonally
- Wires have different lengths (why?):
  - Horizontal: Single, Dual, Quad, Long (12)
  - Vertical: Single, Dual, Hex, Long (18)
  - Diagonal: Single, Dual, Hex



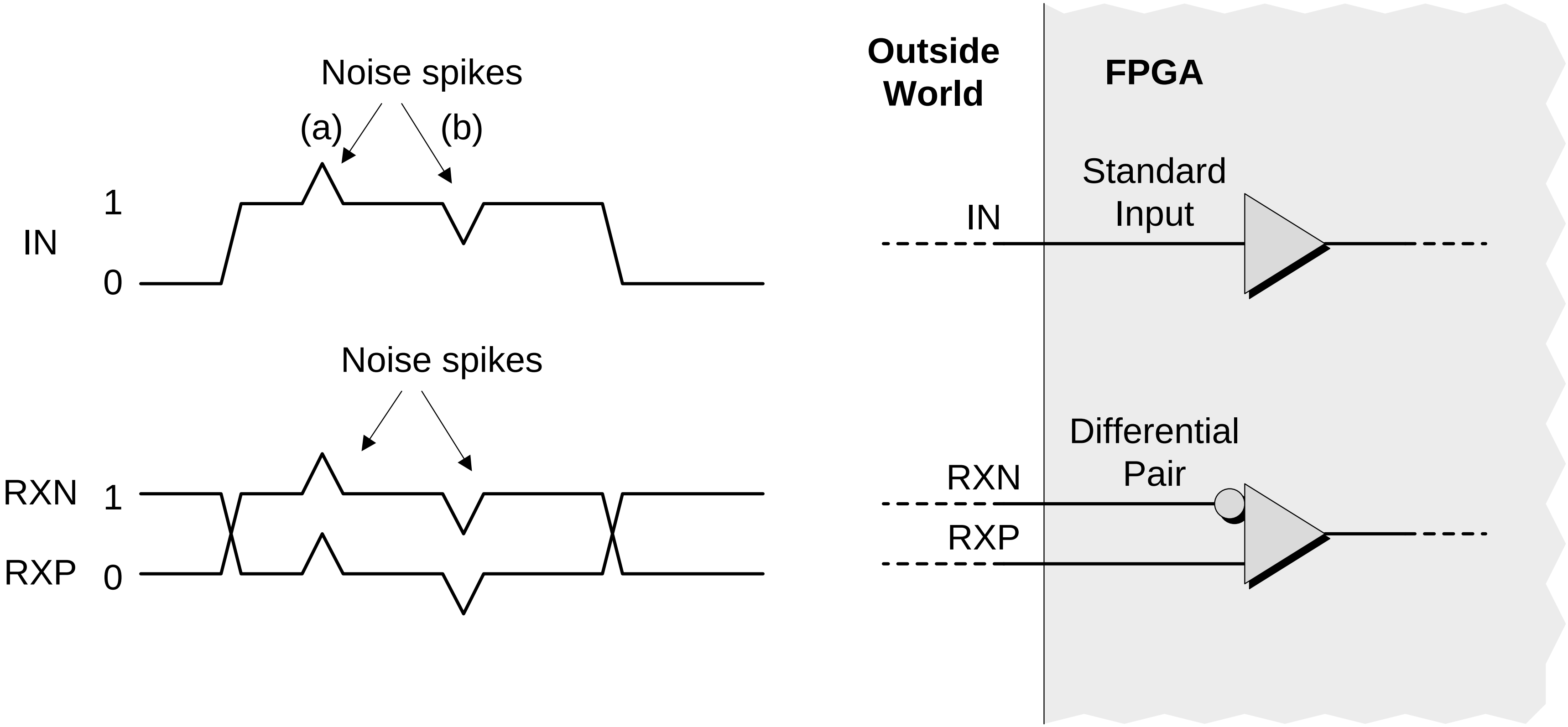
Gigabit Transceiver



- Modern FPGAs have dedicated Transceiver blocks for high-speed communication

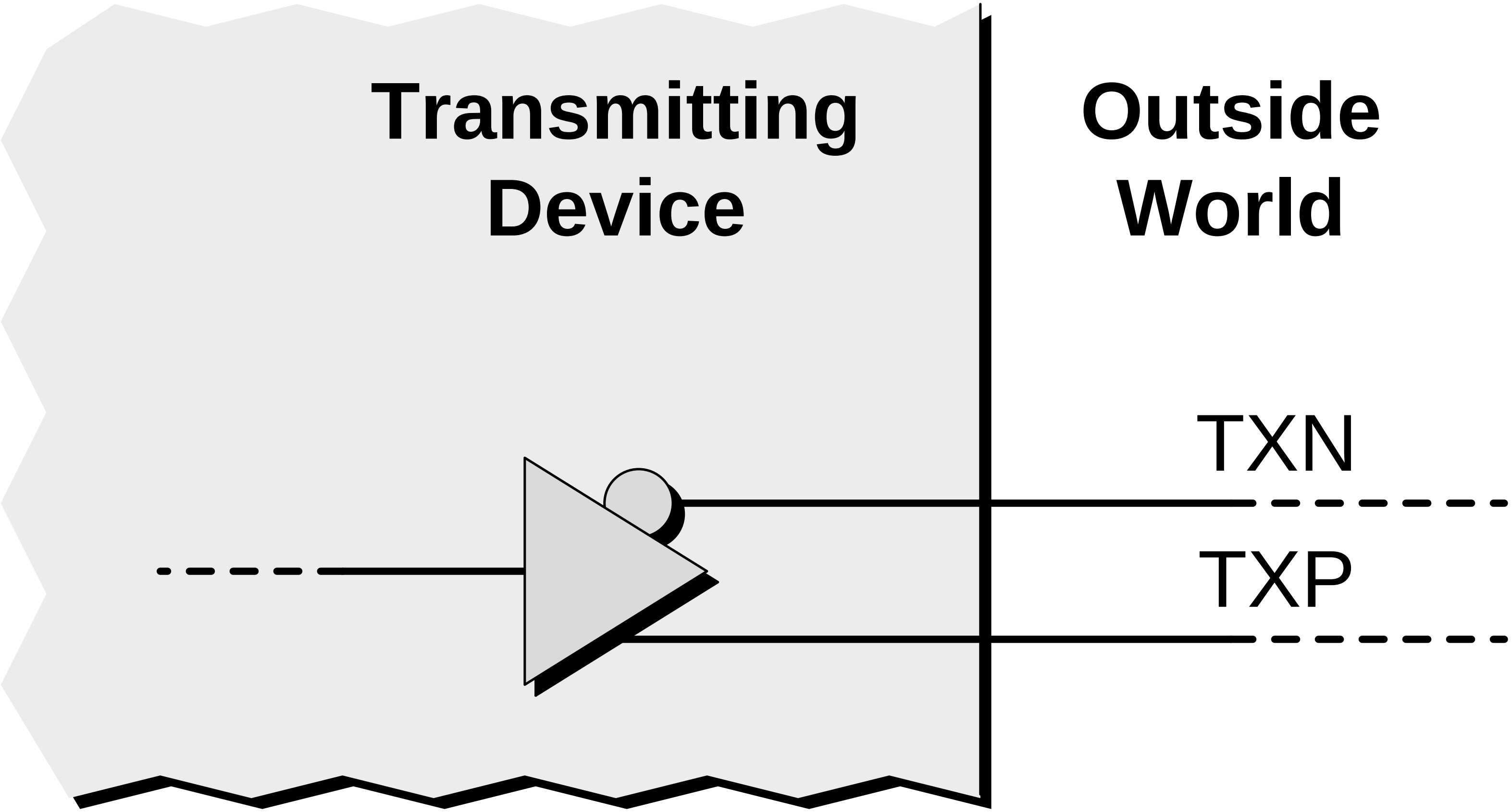
Source: The Design Warrior's Guide to FPGAs Devices, Tools, and Flows. ISBN 0750676043

Why differential signalling?



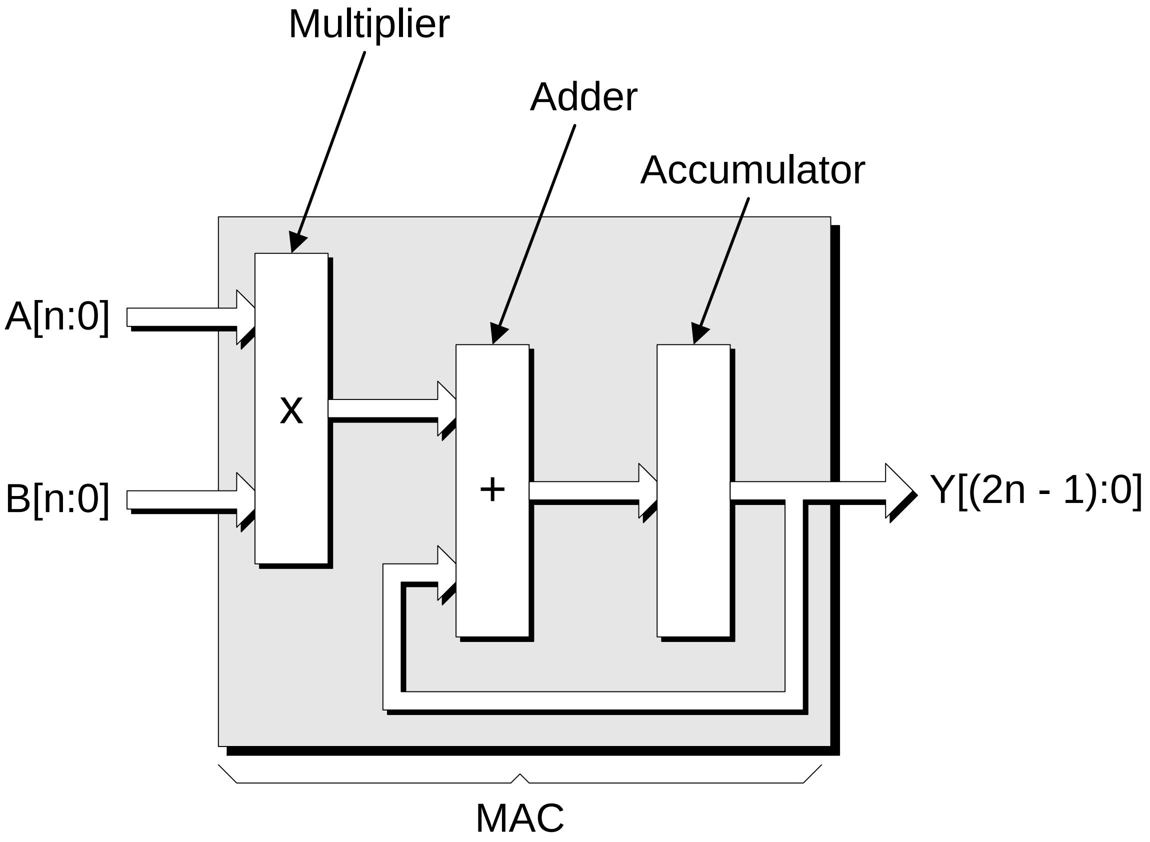
Source: The Design Warrior's Guide to FPGAs Devices, Tools, and Flows. ISBN 0750676043

Generating a Differential Pair



Source: The Design Warrior's Guide to FPGAs Devices, Tools, and Flows. ISBN 0750676043

DSP block



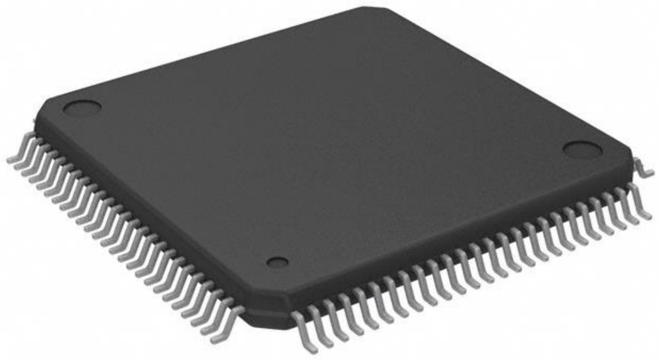
- In 7-series, we have Multiply and accumulate (MAC), instead of simple multiplier

I/O pins

- In addition to more CLBs, Artix family has more I/O pins
- Most Spartan 3 FPGAs have QFP package(or some variant of it)
- Modern FPGA have BGA package(or some variant of it)
- What is BGA? QFP?

QFP

- Quad Flat Package
- Pins are on each of the four sides of the chip
- Limited number of pins, but easier to mount on board



BGA

- Ball Grid Array
- Pins are placed on the whole bottom surface of the device, instead of just the perimeter.
- More pins, but hard to solder and mount on board