Parallel and Reconfigurable VLSI Computing (3)

FPGA Architecture

Hiroki Nakahara

Tokyo Institute of Technology

Outline

- FPGA Architecture Overview
- Programming Technology
- LUT Architecture

FPGA Architecture Overview

FPGA Architecture

<ins>Logic Block (LB)</ins>

- Realize boolean netlist
  Xilinx … Configurable Logic Block (CLB)
  Intel … Logic Array Block (LAB)
  Logic Array Module (LAM)

<ins>I/O element (IOB)</ins>

- Connecting outer resources

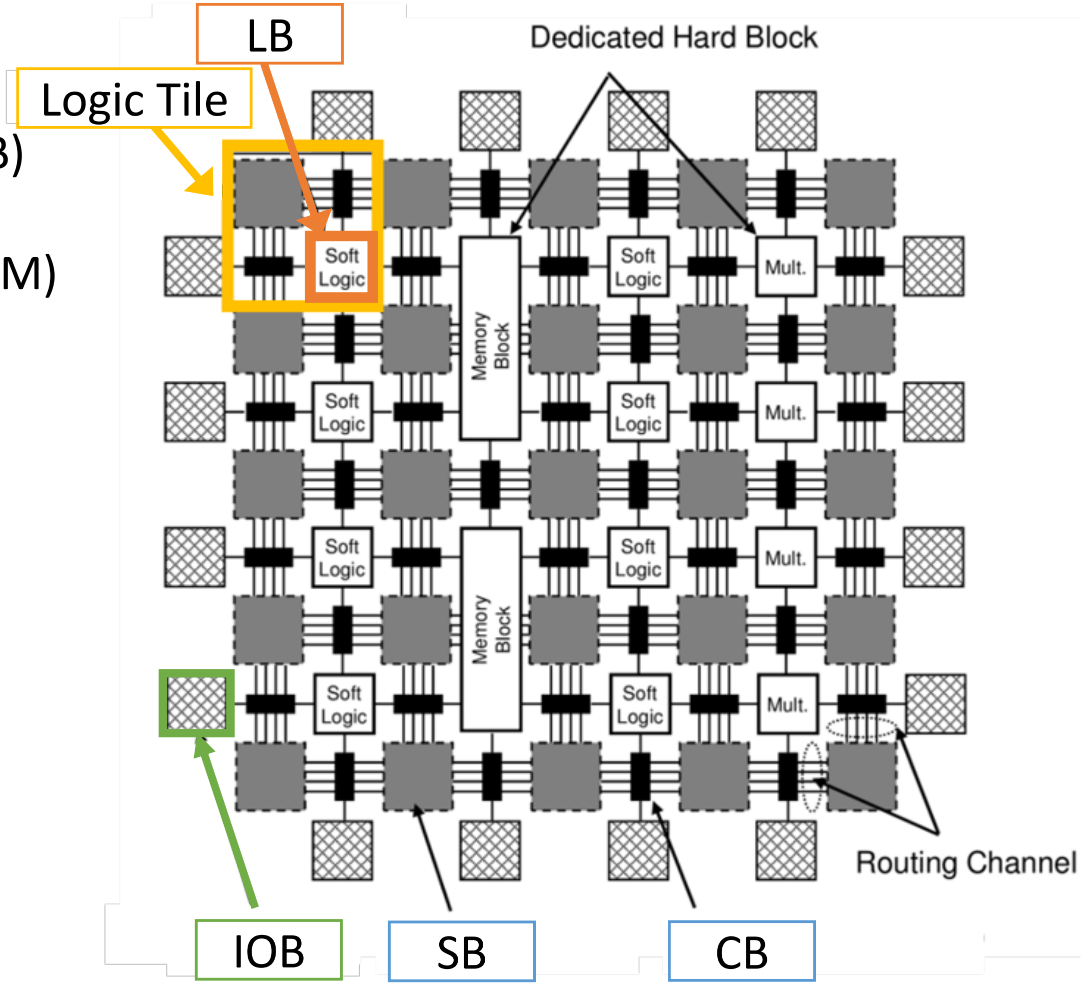
<ins>Routing channel</ins>

<ins>(Connection Block, Switch Block)</ins>

- Connecting FPGA elements

Dedicated Hard Block:

・DSP block
・Memory block
・PLL / DLL



Look-Up Table (LUT)

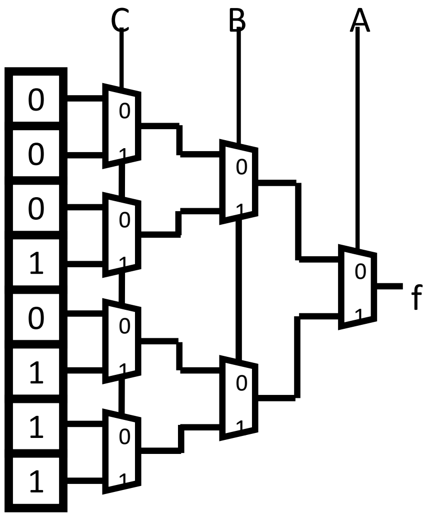
Create truth table according to the number of inputs of LUT

And writes the function value (column of f) as it is in the configuration memory

When the logic function to be realized has more variables (literals) than the number of inputs of the LUT

Implementation using multiple LUTs (described in detail in Chapter 5)

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Example of 3-input majority circuit

Routing Channel

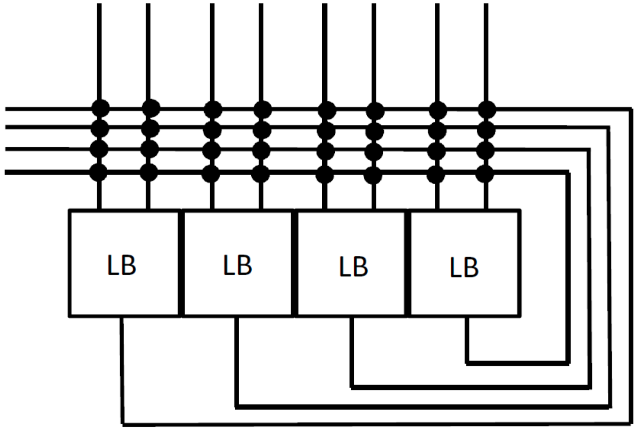
The wiring structure is largely classified into the following 4 types
· Full crossover type
· One-dimensional array type
· Two-dimensional array type (island type)
· Hierarchical type

Classification by logical block and I / O block connection method

It consists of wiring track and programmable switch

Determining the wiring route according
to the value of the configuration memory

Carry In



Carry Out

Routing Variations

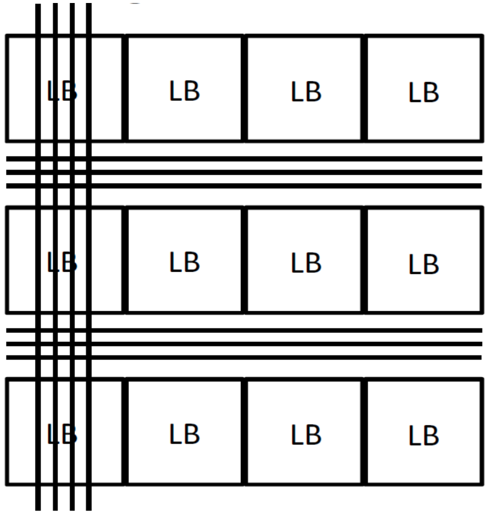
1D Array

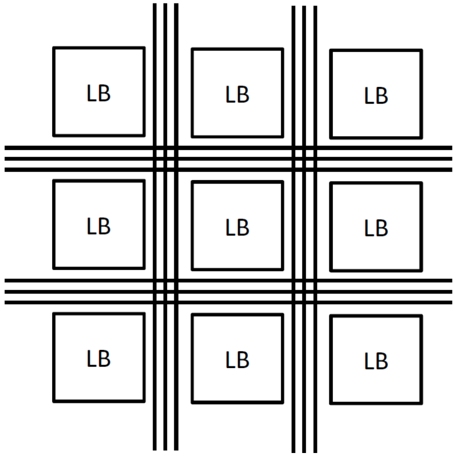
ACT Series FPGA

- Logical blocks arranged in rows, wiring channels arranged in row direction
- Channel is connected with feedthrough wiring
- The number of switches tends to increase

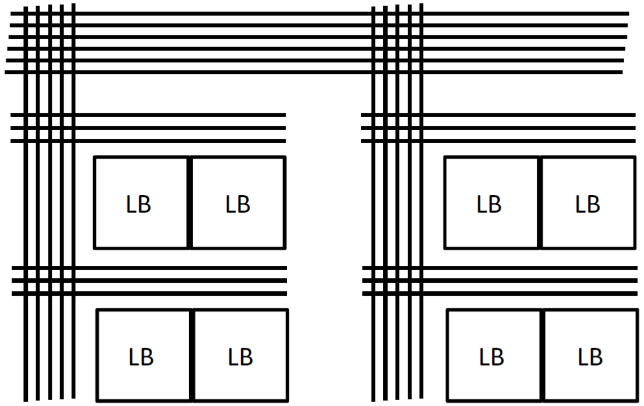
→ Since SRAM type switches are mainstream, area overhead is large

Feed-through





Island type



hierarchy type

Global Routing Architecture

The wiring structure of FPGA is classified into global routing architecture and local routing architecture

Global wiring architecture
Meta viewpoint that does not take into consideration the switch level,
such as connection between logical blocks and the number of tracks per channel

Detailed wiring architecture
Determine concrete connection up to switch arrangement
between logical block and wiring channel

[1]Hierarchical FPGA
- Intel FPGA

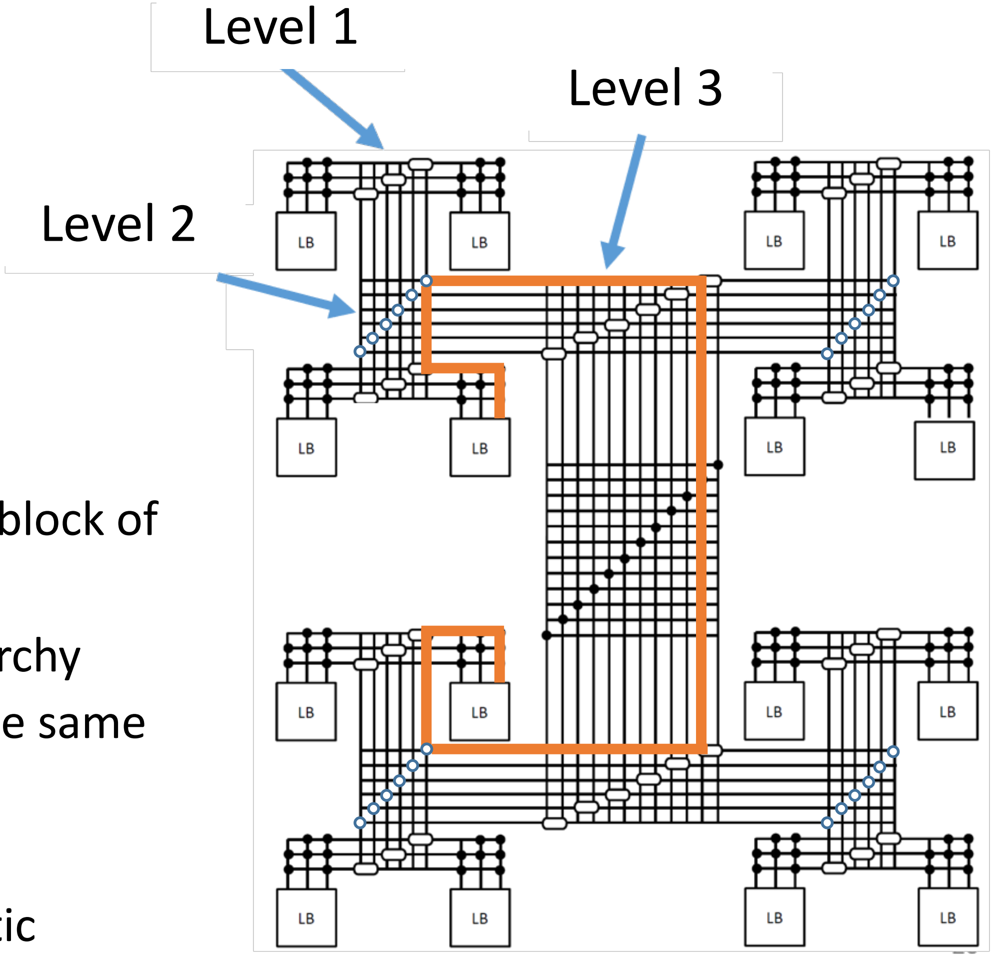
UCB HSRA(High-Speed, Hierarchical Synchronous Reconfigurable Array)
- Three-level structure
- The number of tracks per channel is higher as the upper layer

Global Routing Architecture

In the lower hierarchy, wiring between a plurality of logical blocks is performed
Reduced switches required for signal propagation within the same hierarchy
→ High-speed operation is possible

In the case of a circuit not conforming to the hierarchical type, the usage rate of the logical block of each hierarchy extremely decreases
Increase delay penalty once crossing the hierarchy
Delay increases if not physically close but at the same hierarchical level

Variations in parasitic capacitances and parasitic resistances in recent processes can cause variations in delay even within the same layer



Global Routing Architecture

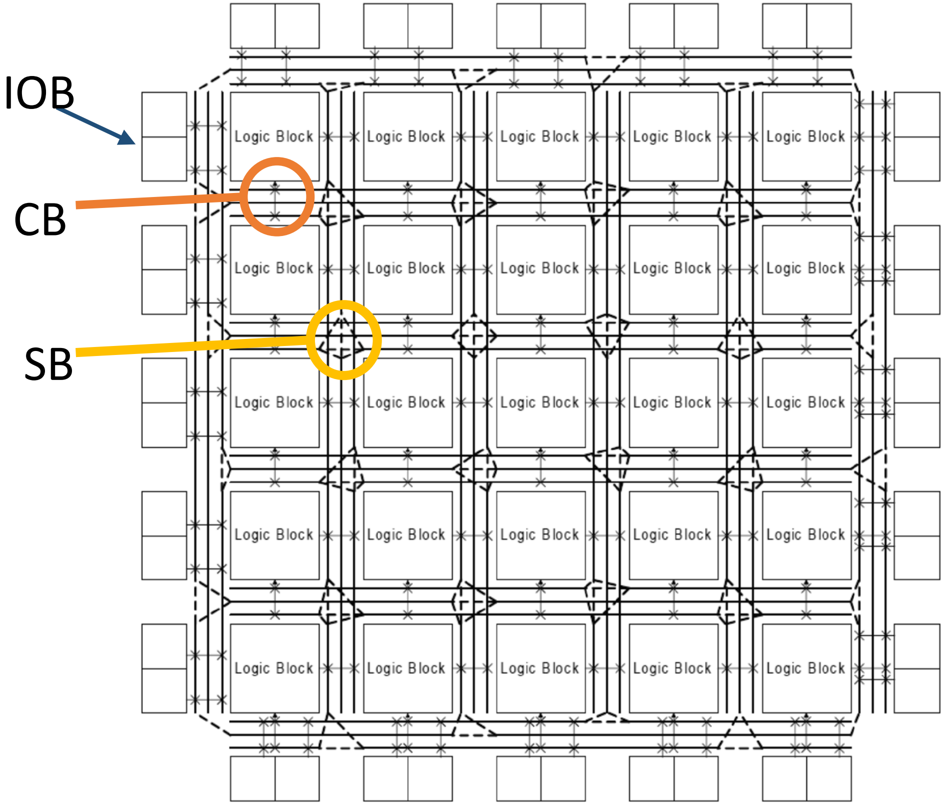
[2] Island style FPGA

・Xilinx FPGA

· Vertical and horizontal wiring channels exist between logical blocks

- The connection between the logic block and the wiring block is performed in two directions, or in four directions

· If the logic tiles are made uniform, the mounting time in the wiring process decreases



Local Routing Architecture

Switch placement between the logical block and the wiring channel, and wiring segment length are determined

W : # of tracks per routing channel

Connection block (CB) has an input/output

F_{cin} : Flexibility for input CB

(# of connections with input CB/W)

F_{cout} : Flexibility for output CB

(# of connections with output CB/W)

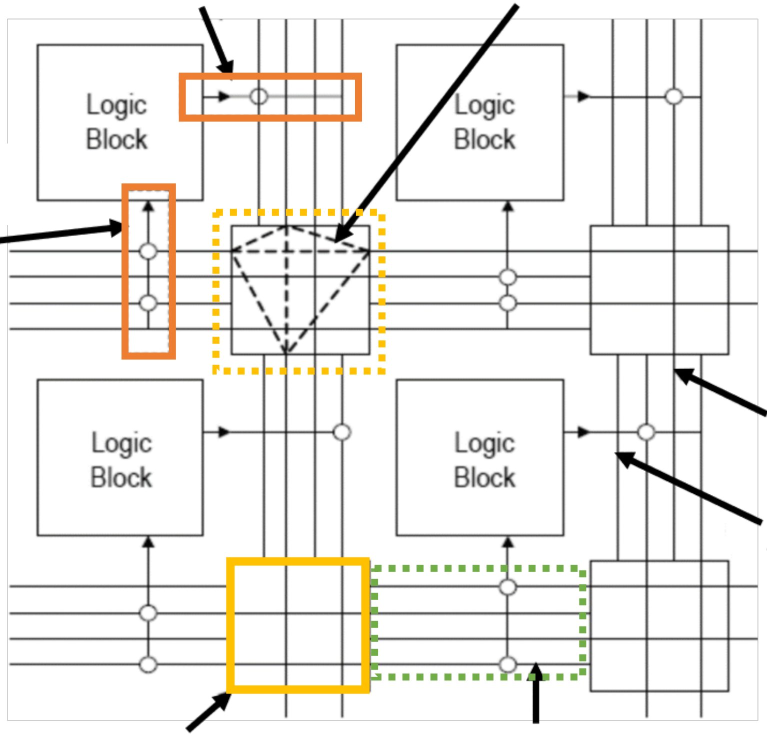
The switch block (SB) exists at the intersection of the routing channels in the horizontal direction and the vertical direction

F_s : SB Flexibility

ex) F_{cin} = 2/4 = 0.5

F_{cout} = 1/4 = 0.25

F_s = 3 (3 input and an output)



Input CB

Programmable Routing switch

Input CB

Long Wire Segment

Short Wire Segment

SB

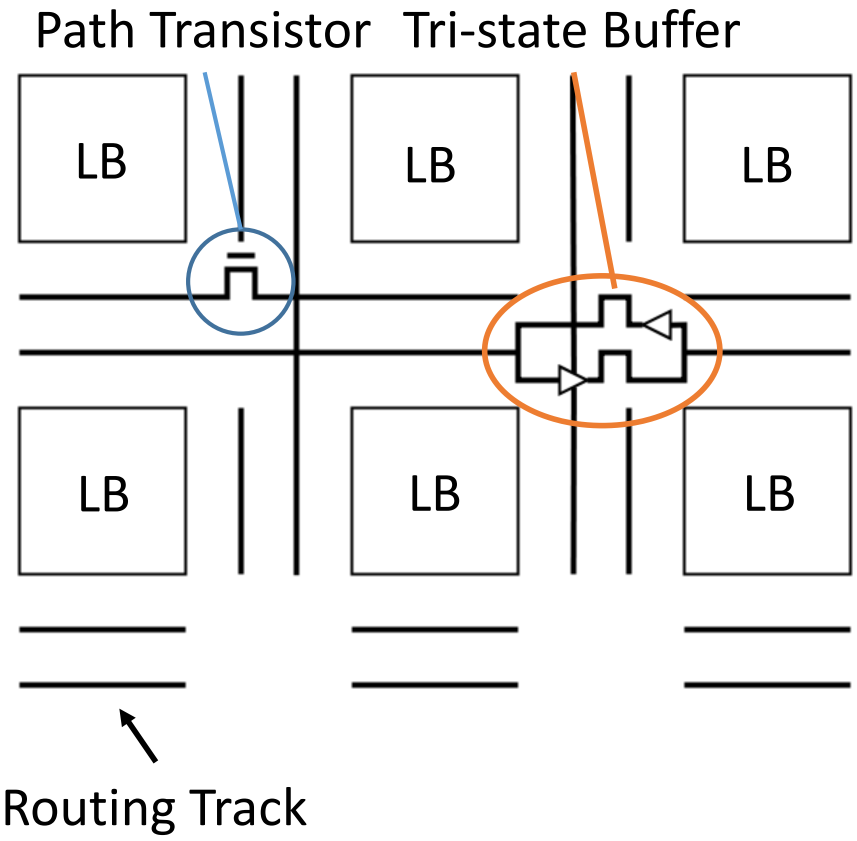
Routing Track

Detail Architecture for Connection

The structure of the programmable switch is important for determining the wiring architecture of the FPGA
- Pass transistors and tri-state buffers mixed in many FPGAs
Pass transistor: transistor switch that can freely control ON / OFF
Tri-state buffer: A circuit that can take three states of H level, L level, and high impedance

Pass transistors
Less number of switches with short path, but repeaters are necessary if you pass many steps
Tristate buffer
Suitable for long paths

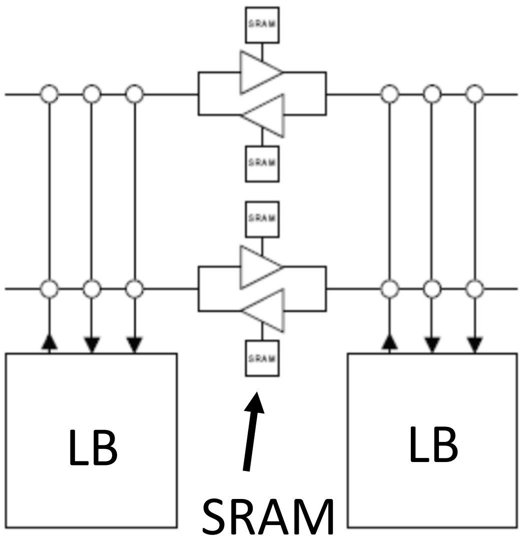
Achieving good performance when these ratios are halved



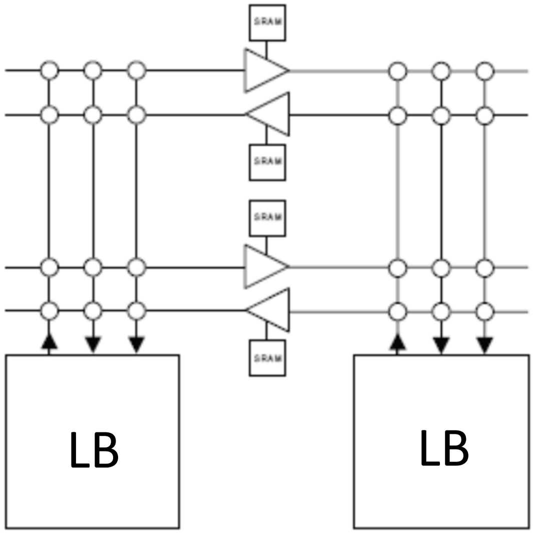
Channel Track Direction

Bidirectional wiring
- Reduce the number of wiring tracks
- One switch is not used
- Impact on delay due to increased wiring capacity

Unidirectional wiring
- Twice as many tracks as bidirectional wiring
- Switch always used
- Small wiring capacity



(a) Bidirectional track



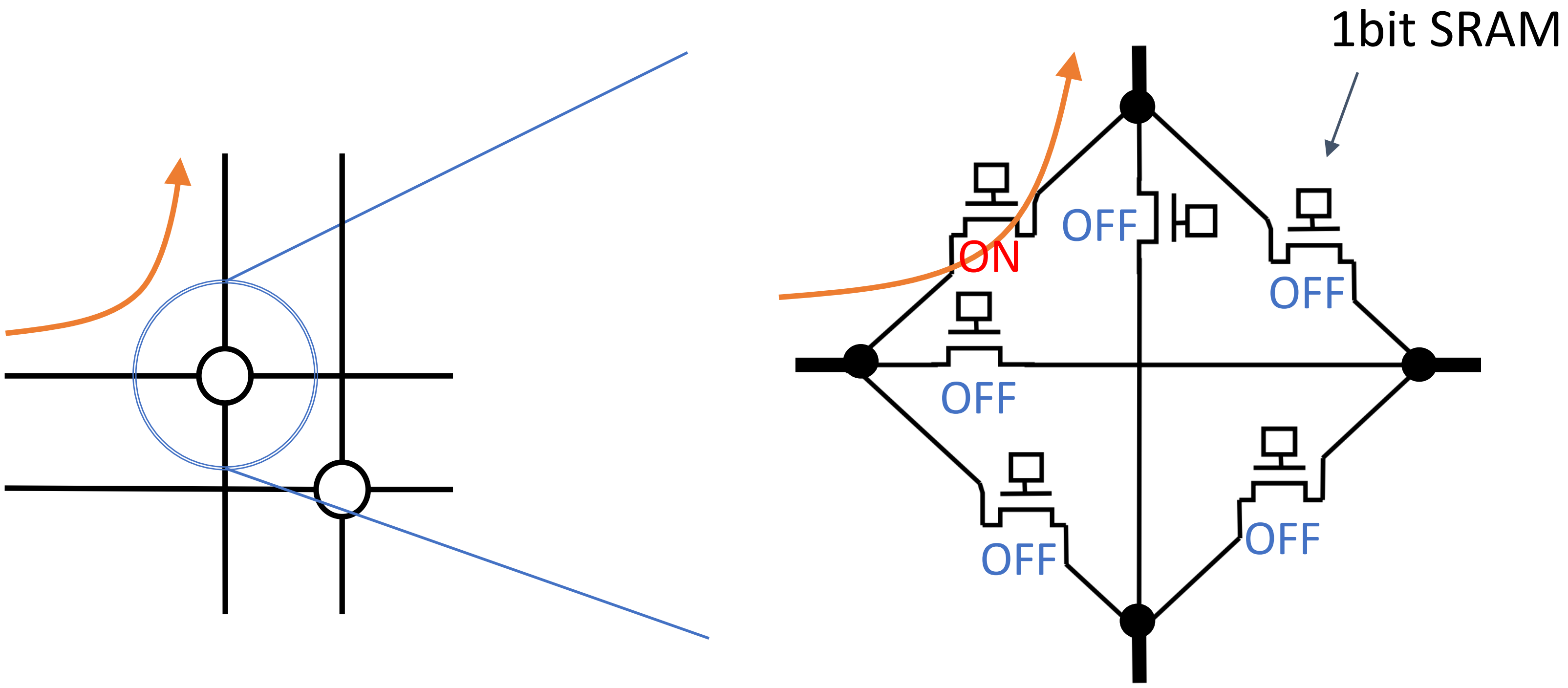
(b) Unidirectional track

There is a performance trade-off as described above
In recent years, the number of metal layers of the transistor has increased, and from the ease of design it has shifted to unidirectional wiring

Switch Block

SB is positioned at a point where the wiring channels in the horizontal direction and the wiring direction in the vertical direction cross each other, and the wiring route is determined by the programmable switch

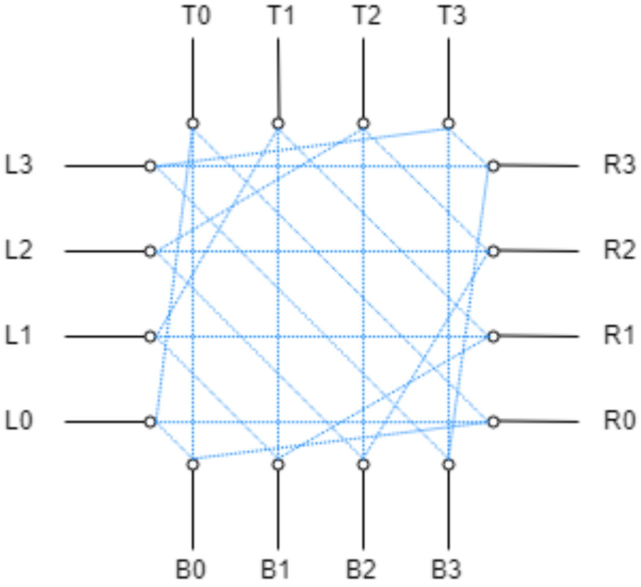
There are three kinds of topologies: Disjoint type, Universal type, and Wilton type



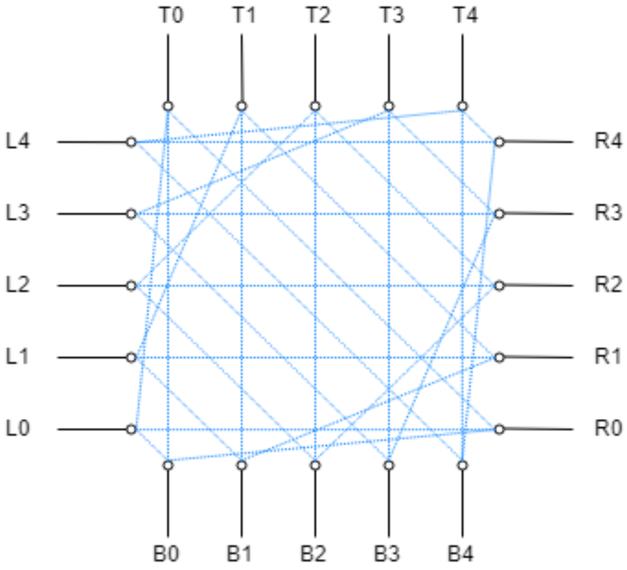
Switch Block with 6 path transistors

(1) Disjoint Type

It is also used in Xilinx's XC 4000 series and so it is also called Xilinx type SB
L0 is connected to T0, R0, B0
Since the connection is realized by six switches, the total number of switches is W=6
Connect up and down, diagonally same number
The number of switches can be reduced, but it can only be connected with tracks of the same value
→ Flexibility of routing is low



W = 4

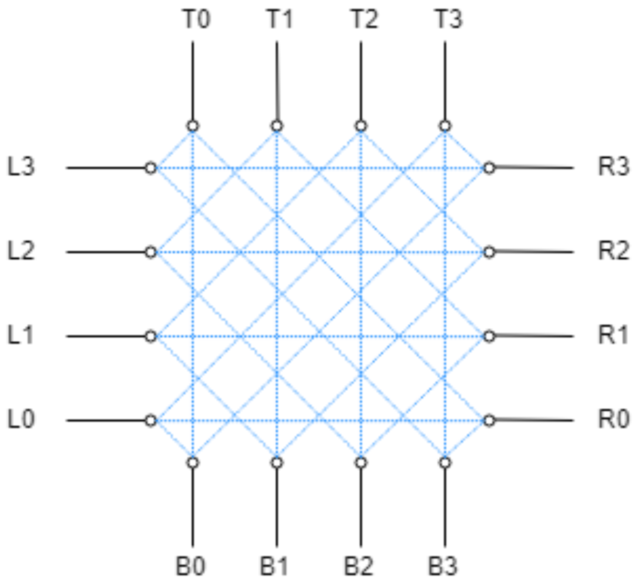


W = 5

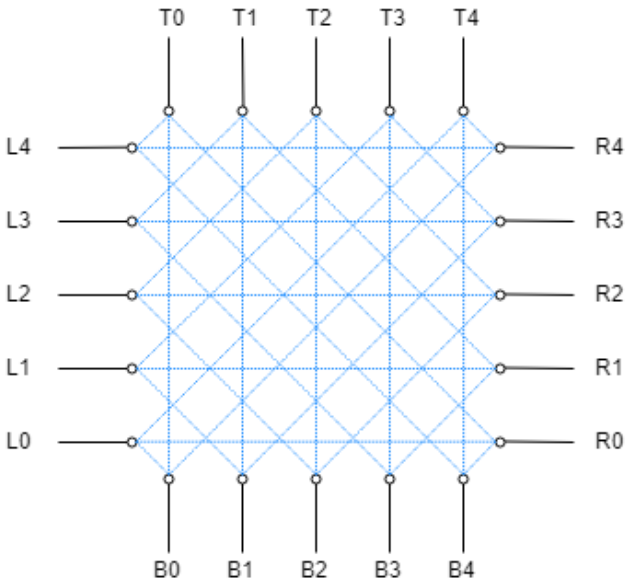
(2) Universal Type

Like the Disjoint type, it consists of the number of switches of 6 W
Connect between pairs of two tracks
Tracks with no pairs like W = 5 are the same as Disjoint type

Compared to the Disjoint type, the total number of tracks can be reduced
Suppose only a single line, not support other wiring lengths



W = 4

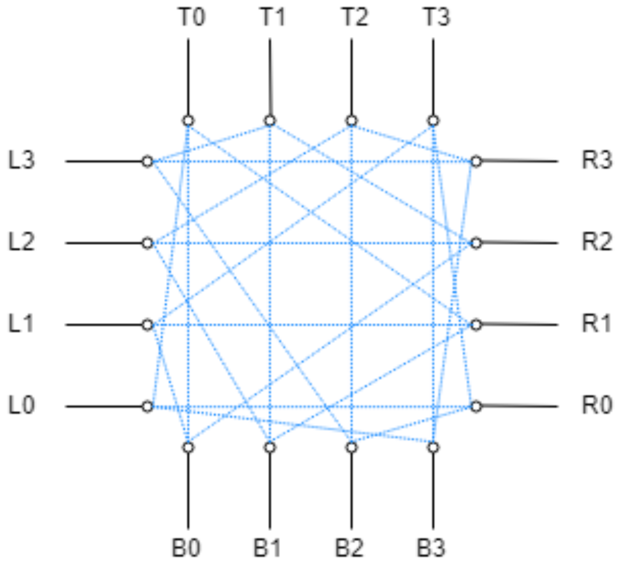


W = 5

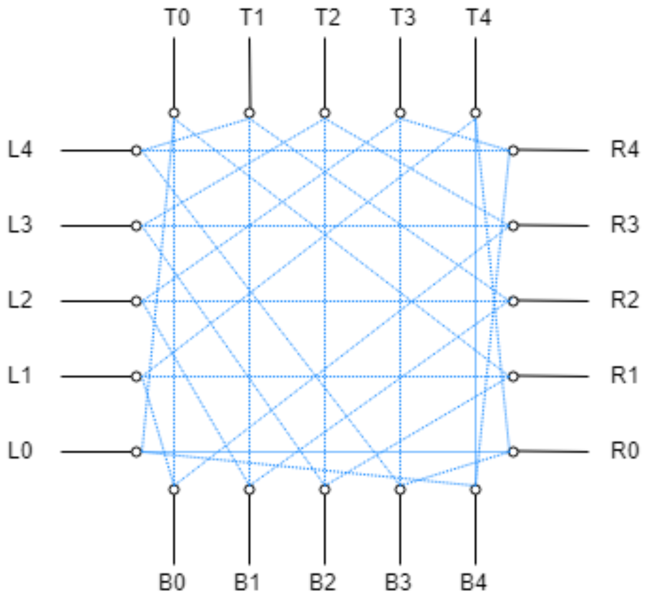
(3) Wilton Type

Wired tracks with different values can be connected with a W=6 switch
High degree of freedom of wiring compared to other topologies
 - Can be wired to tracks with different numbers in one SB

Circuit can be constituted by clockwise and counterclockwise wiring
Efficiency improvement of manufacturing test of FPGA



W = 4



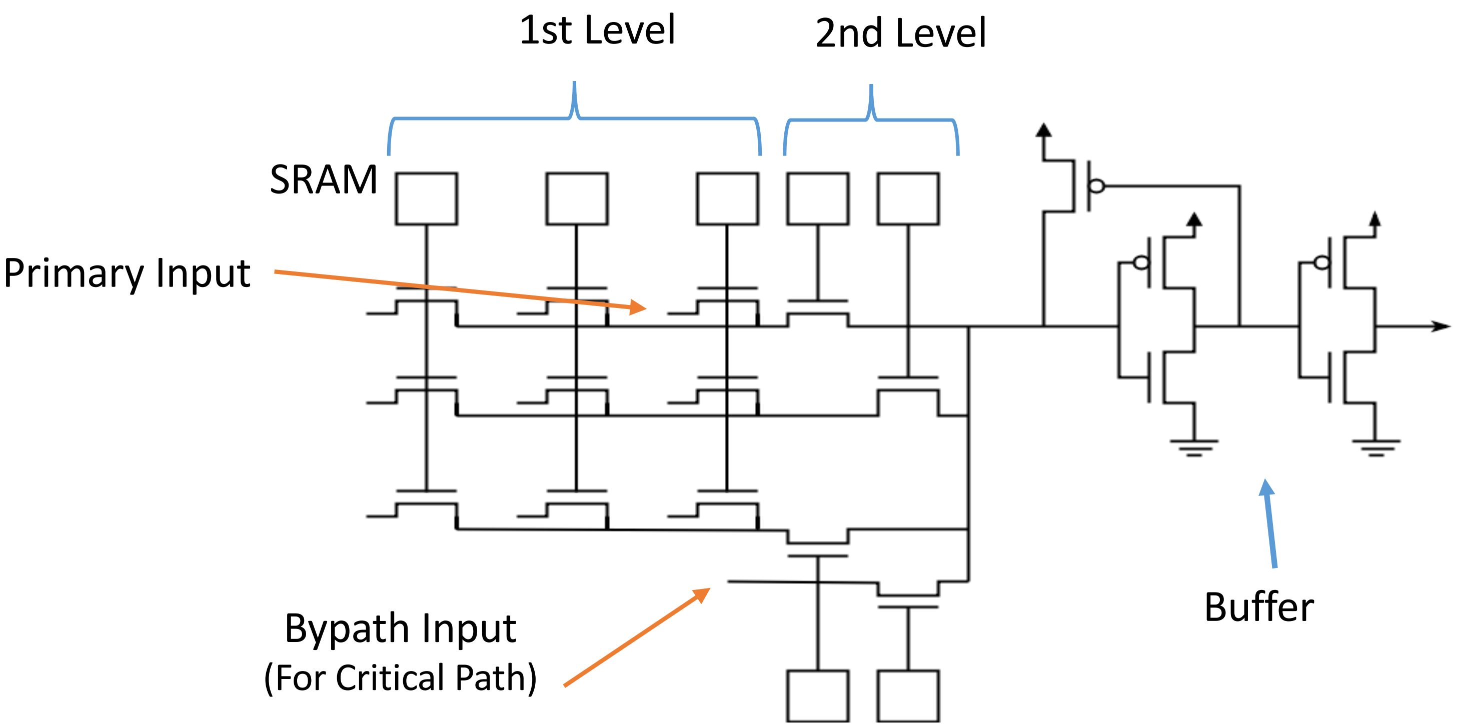
W = 5

Multiplexer

Circuit programmable switch big transmission delay resulting multiplexer many select a large number of signal significantly to operate delay unidirectional wiring is effects - especially

Circuit structure for multi-input multiplexer for Intel (Altera) Corp. Straitx II (9-input + an additional input for a critical path signal)

To reduce the path delay even if memory size increases



I/O Block (IOB)

Input and output elements are configured with input and output dedicated module, it realizes the connection of I/O pin devices and logical block between the interface (I/O block)

- In addition to dedicated pins for power supply and clock, FPGA I/O pins have user I/O which the user determines input / output polarity

The I/O block,

· Input/output buffer

· Output driver

· Polarity specification

· High impedance control

Such as a logic block in the FPGA and the I/O pad of the device to exchange input and output signals

I/O pins have functions such as power supply, clock, user I/O etc.

I/O Block (IOB)

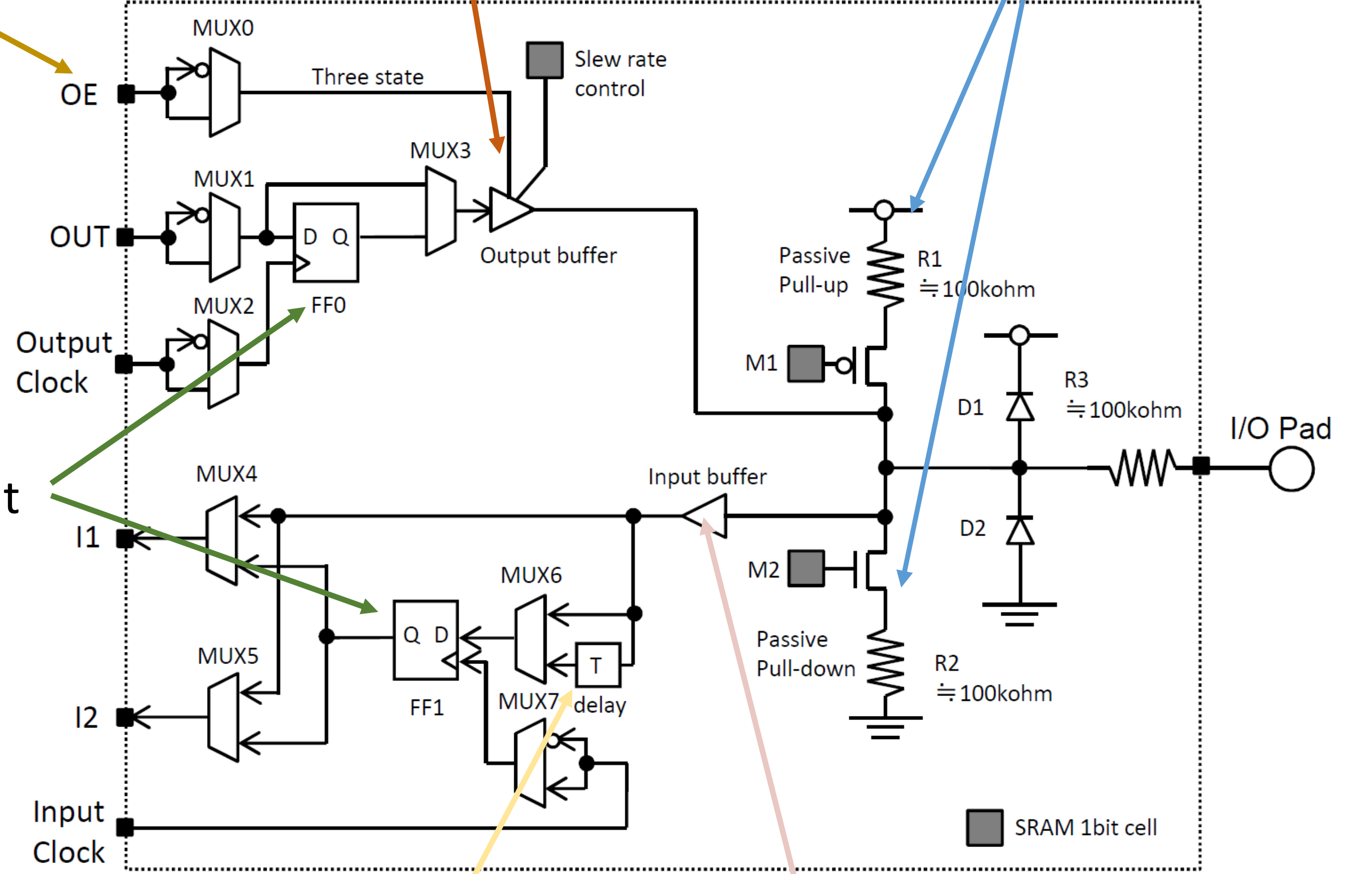
(for Xilinx XC4000)

Control of output buffer with output enable signal (OE)

Control slew rate of output buffer

Pull-up/pull-down resistors can be connected to the output section
Fix device output to 0 or 1

Each I / O has FF
Latency adjustment possible



In order to guarantee the hold time, a delay circuit is provided at the input stage of the MUX 6

Input buffer has TTL or CMOS threshold

Hard Macro

Along with the increase in the scale of FPGAs, it is better in terms of performance and mounting efficiency to realize general-purpose interfaces commonly used by many systems as dedicated hardware than user circuits
→ Such a dedicated hardware circuit is called a hard macro

Interface for a hard macro

Examples of hardware macros other than hardware multipliers and DSP blocks
· PCI Express interface
· High-speed serial interface
· External DRAM interface
· Analog / digital converter

Only one or a few are prepared on each FPGA
→ There is a fear that the wiring may become longer unless the wiring route is taken into account

DSP Block

Early FPGAs are based on LUTs and programmable wiring elements
Realize desired logic circuit with interconnection

⇩

The chip size of the FPGA increases and the leading role of the main target application shifts to digital signal processing (DSP) such as FIR filter and fast Fourier transform (FFT)
→ The computing unit for implementing multiplication is important
Multiple logical block connections are required for LUT-based multiplier configuration
→ difficult to achieve high computing performance
Dedicated circuit of multiplier is implemented on FPGA as hardware block

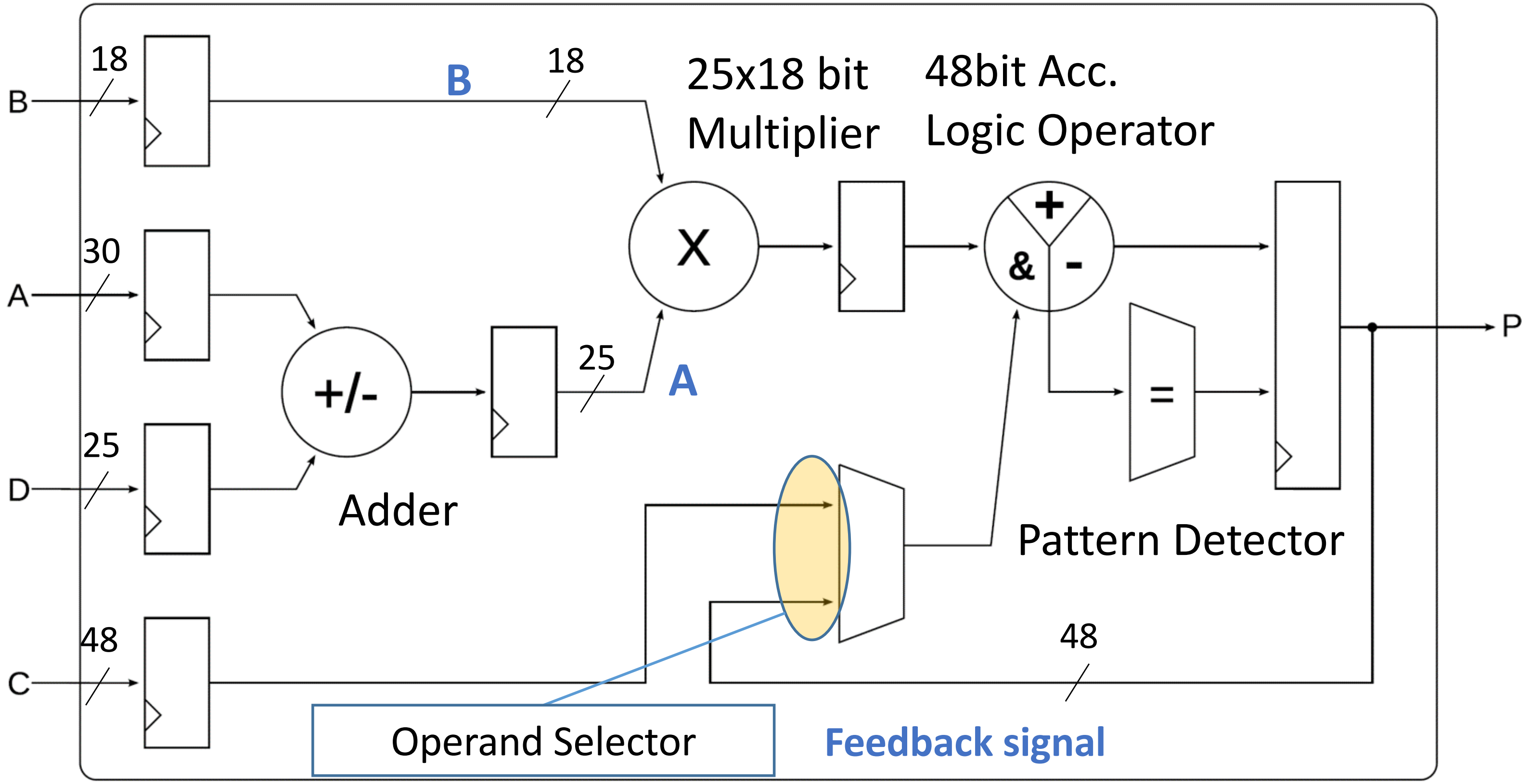
Because it is a dedicated computing circuit ...

Increase system throughput ⇔ **Low flexibility**

DSP Block for Xilinx 7 series FPGA

DSP48E1 slice configuration adopted in Xilinx 7 series architecture

- Spartan-7, Kintex®-7, Artix®-7, Virtex®-7



MAC (Multiply-Accumulate) Operation $\underline{Y = A \times B + Y}$

Embedded Memory

(Block Memory: BRAM)

· Early FPGA architecture
User circuit realized using only LUT and FF based logic block
→ Memory elements are FF only

Since it is not possible to store a large amount of data in the chip, it is necessary to connect an external memory
→ The bandwidth of the connection becomes a bottleneck

· Recent FPGA architecture
Efficiently realize memory element inside chip
→ Embedded memory
Embedded memory is roughly divided into two types
- Memory block as hard macro, LUT in logical block

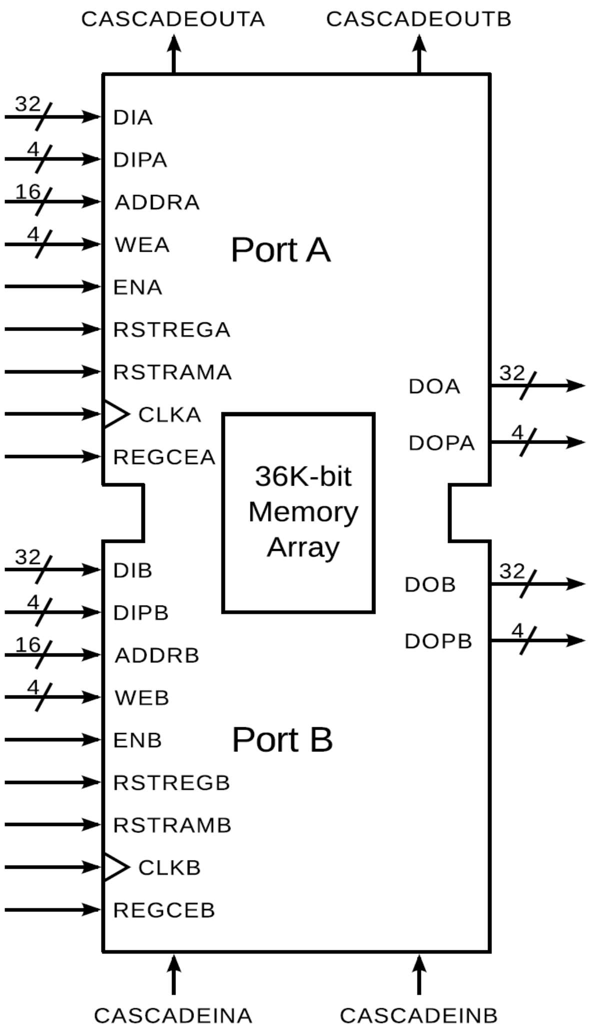
BRAM Hard Macro

Introduced memory block as hard macro into FPGA
- Xilinx FPGA architecture is called Block RAM (BRAM)

· One BRAM can be used as one 36K bit memory
  or two independent 18K bit memory
- One 72K bit memory can be configured
  by combining two BRAMs

· FIFO (First-In-First-Out) memory for data transfer
  between submodules can also be configured
  by using it as dual port memory of A and B

→ However, asynchronous access is not possible



Hard Macro Processor

A microprocessor is indispensable when realizing a complex system
A processor can be configured as a user circuit from the advantage
that a general-purpose circuit can be realized on an FPGA
→ Soft core processor
A processor made as a hard macro has better performance
→ Hard core processor

Both Xilinx and Intel (Altera) FPGAs embedded the ARM processor

Xilinx Zynq-7000 EPP

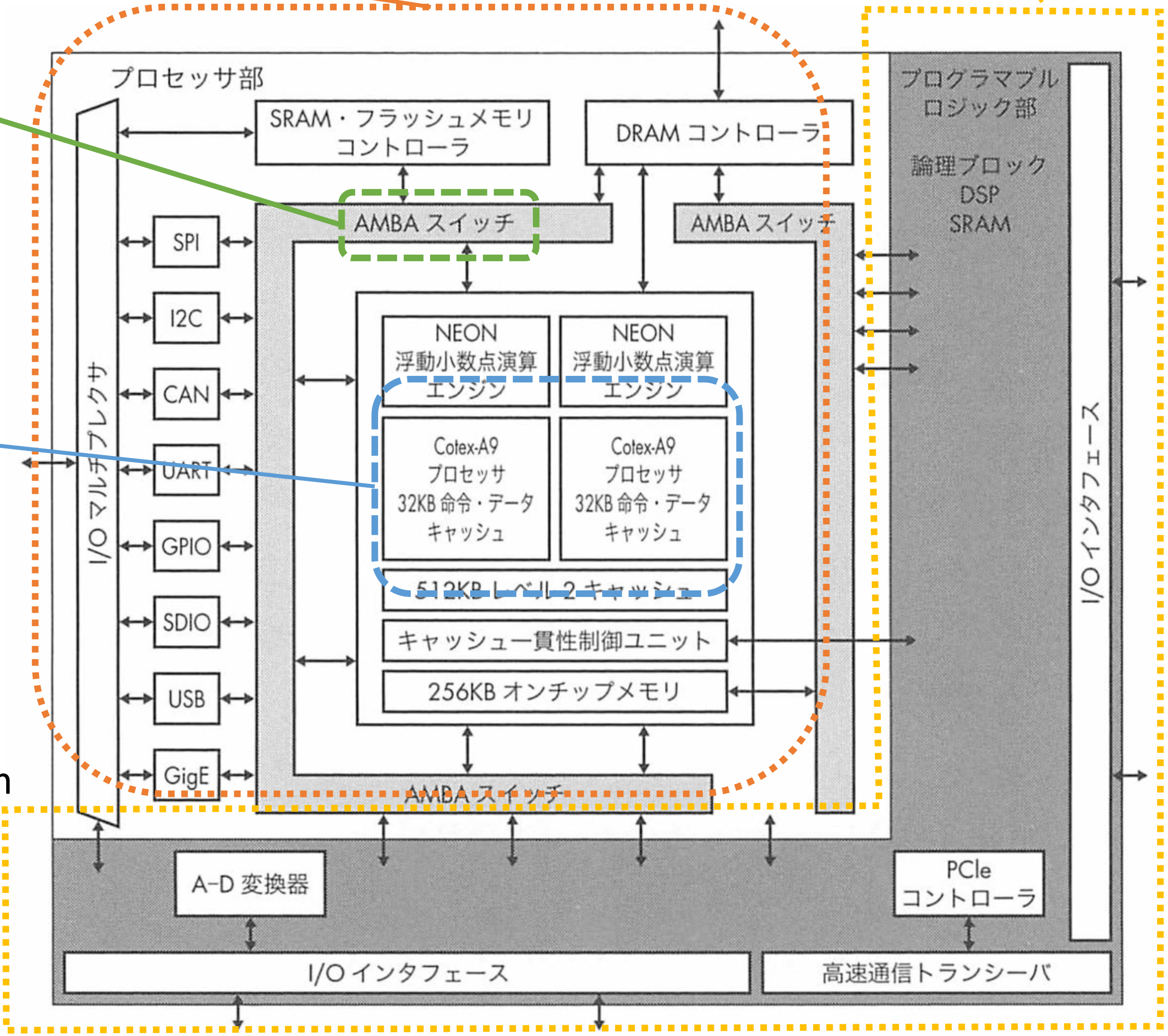
Processing System (PS)

Programmable Logic (PL)

AMBA protocol (On-chip interconnect)

Multi-core

Offload part of the processor to custom user hardware



PLL and DLL

The operating frequency on the FPGA differs according to the critical path of each circuit

↓

· Clock signals with various frequencies in the chip
· Clock signals with no phase difference with the external input clock so that the circuit on the FPGA communicates with the external system
· Clock signals with different frequencies and phase differences according to the external interface

↓

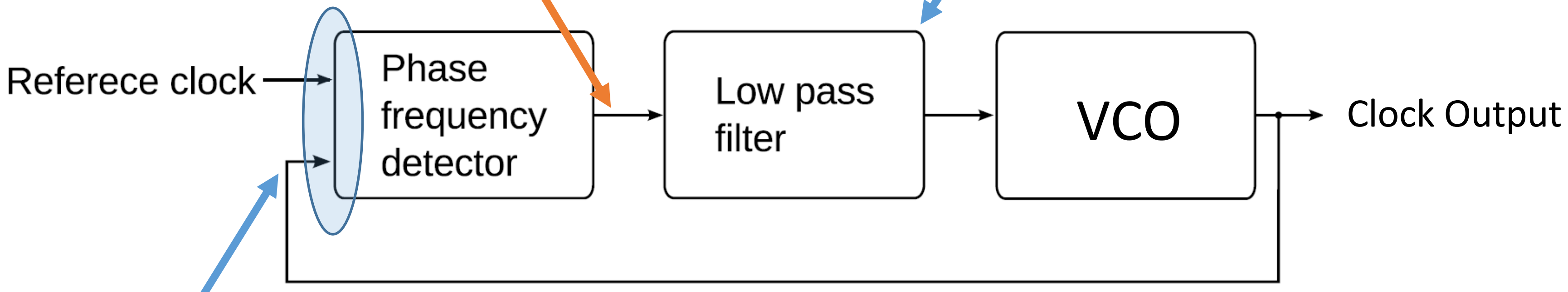
Programmable PLL (Phase-Locked Loop) that can generate various clock signals based on the external reference clock

PLL Configuration

Voltage-controlled oscillator VCO (Voltage-Controlled Oscillator)
An oscillator is the central part of the clock generation and changes the frequency
Convert to analog voltage signal using charge pump circuit
- Electronic circuit that raises the voltage by combining a capacitor and a switch

Error Signal

Since the system is not stable when the voltage signal is used as it is, cut high frequency components with LPF



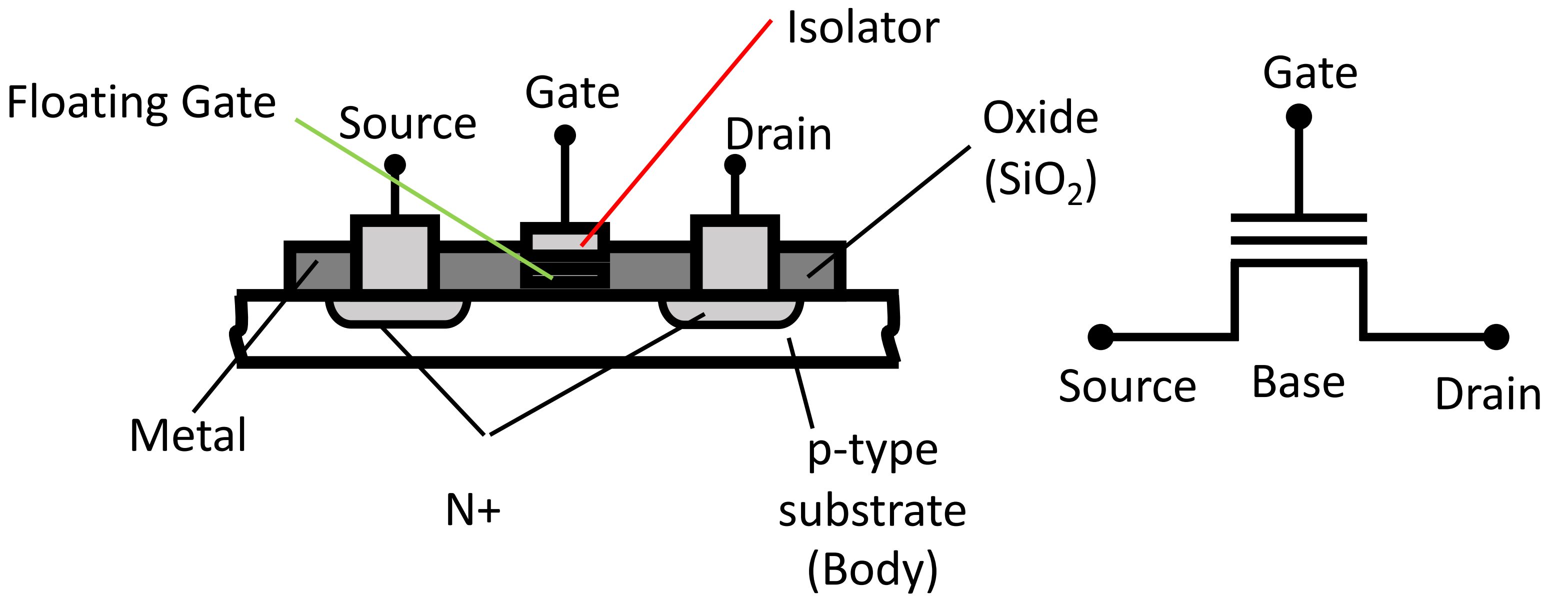
Synchronize phase and frequency

Feedback signal

Programming Technology

Flash Memory

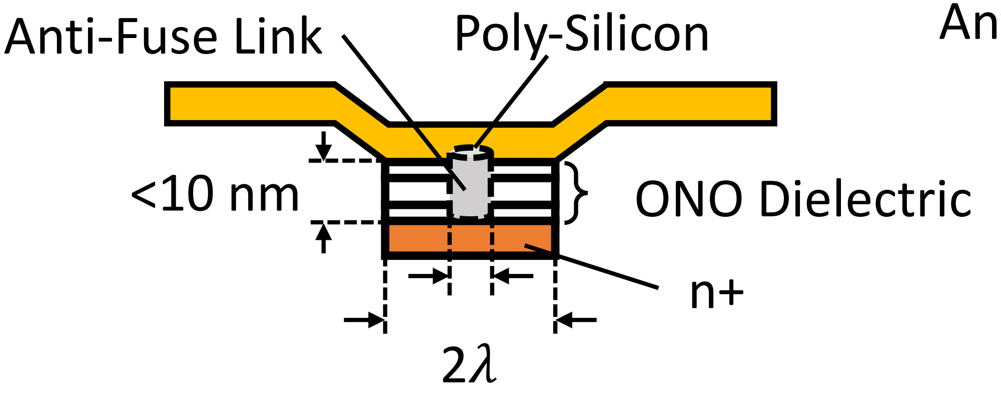
- A type of a nonvolatile memory of EEPROM
  - A floating gate in the insulating film
- The floating gate is formed of a polysilicon film
  - Floating gate electrode in insulator (SiO 2) without connection
- Writing methods
  - NAND type ... Requiring high voltage
  - NOR type ... Requiring high current

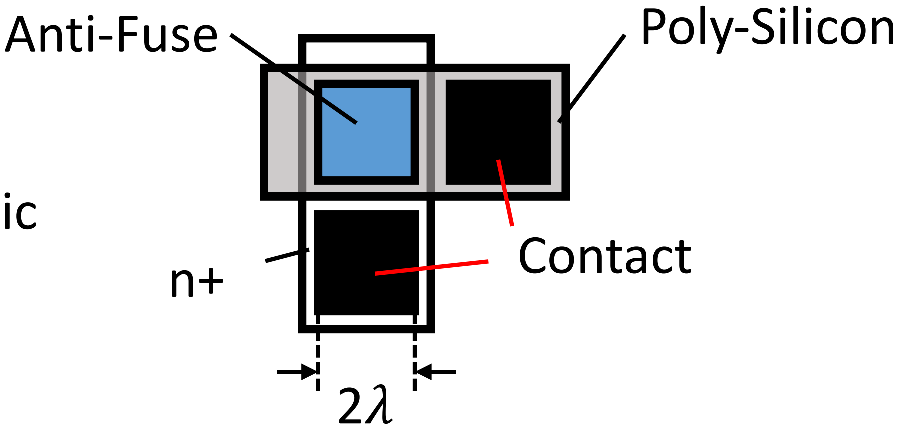


Anti-Fuse

- Typically, it is open (insulated), continuity is burned out by applying current
- Example of Actel PLICE (Programmable Logic Interconnect Circuit Element)
- PLICE uses poly-silicon and n+ layer as a conductor
- Insert Oxide-Nitride-Oxide (ONO; oxide film-nitride film-oxide film) dielectric as insulator
- The ONO dielectric typically has a voltage of about 10 V and a current

  of about 5 mA to connect up and down
- The size of the anti-fuse itself is roughly equivalent to that of the contact hole





Static Memory (SRAM)

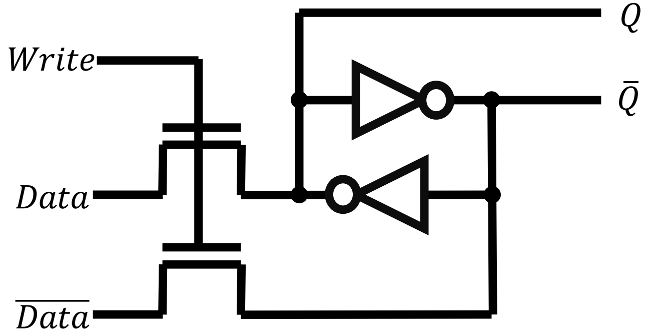
The static memory consists of a flip-flop composed of two CMOS inverters and a past transistor (PT)

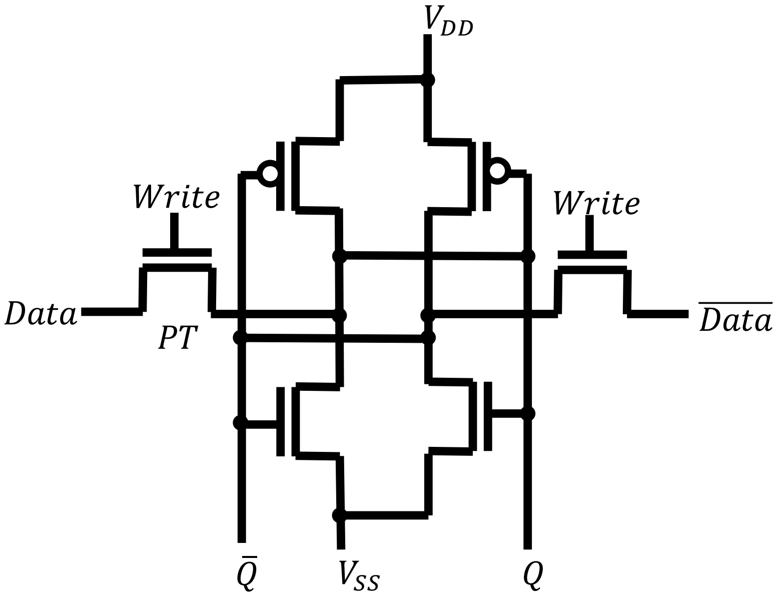
Information is stored in the bi-state (0 and 1) of the flip-flop

Drive the word line (connected to the Write signal) from the address information

The output of the memory cell is amplified by a sense amplifier and output

Since the FPGA always needs to read out, always draw out the output from the flip-flop section(Q / $\bar{Q}$ output)





Comparison with Programming Technologies

	Flash	Anti-Fuse	SRAM
Non-Volatile?	Yes	Yes	No
Reconfigurability?	Yes	No	Yes
Area	Medium	Small	Large
Device Process	Flash process	CMOS Process	CMOS Process
		(Additionally Anti-Fuse one)	
In-circuit Program (ISP)?	Yes	No	Yes
Switch Resistance [Ohm]	500~1000	20~100	500~1000
Switch Capacitance [fF]	1~2	<1	1~2
Programming Yield [%]	100	>90	100
#Programming	About 10000	Only once	Unlimited

LUT Architecture

Decision of LUT(Look-up table) Size

· Area efficiency trade-off

- How efficiently logical blocks are used during circuit implementation

By increasing the functionality of one logical block

Reduce the number of blocks required for desired circuit implementation ⇔ Increase area per logical tile

Especially with respect to the input size k of the LUT

→ As the value of k increases, the number of logical blocks required for mounting decreases

→ Increase area due to 2^k bit configuration memory required

→ The number of input / output pins of the logical block also increases, the area of the wiring also increases

Area = # Logic Blocks × Area per Logic Block

Trade-offs for LUT

Speed (Performance):

By increasing the number of inputs for an LUT

#LUTs on a critical path decrease ⟺ Inner delay for an LUT increase

Observations:

If the input size k of the LUT is increased,
→ The number of logic stages decreases and the operation speed increases
→ Waste occurs when implementing logic functions less than k inputs,
and area efficiency deteriorates

If the input size k of the LUT is reduced,
→ The number of logic stages increases and operation speed deteriorates
→ Improve area efficiency

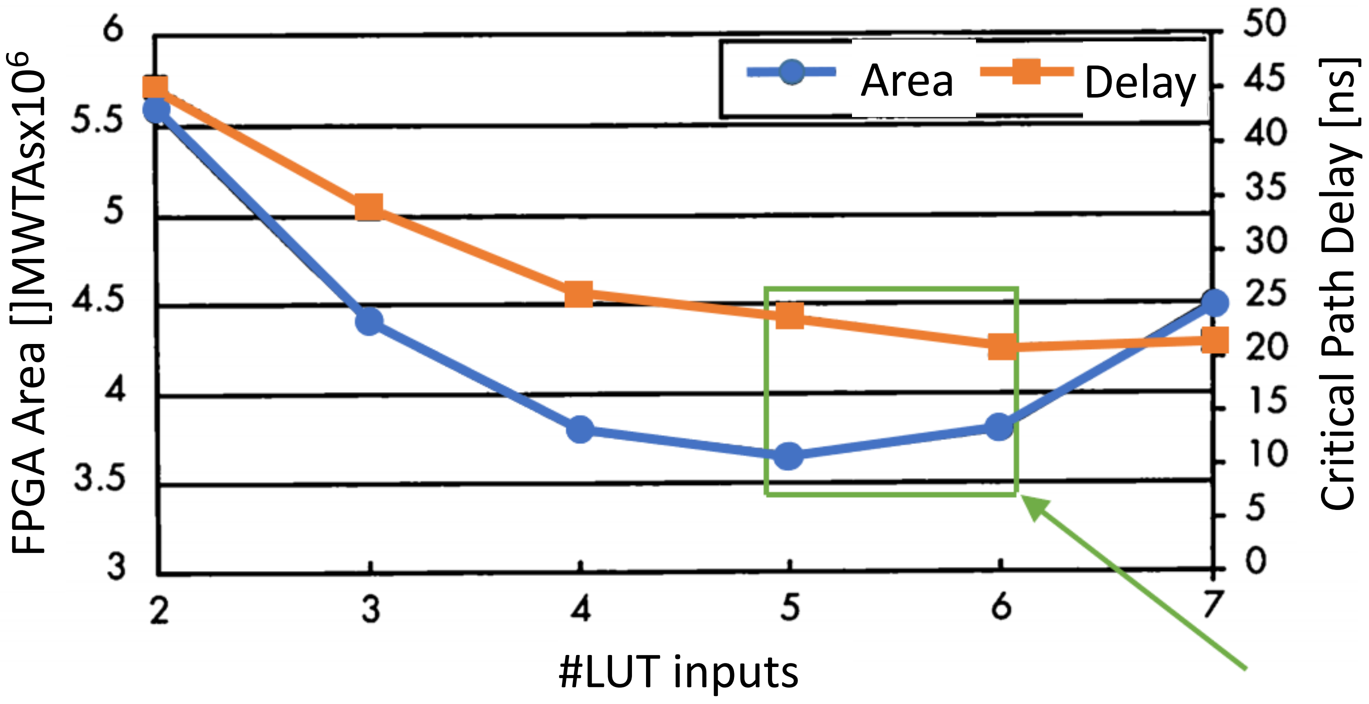
Area vs. Delay for k-input LUTs

Evaluation of logical block architecture

· In the early 1990's, it was judged that the efficiency of the 4-input LUT was good

(for Virtex 4 from Xilinx Corp., 4 inputs for Intel's Stratix)

→ Designed at the transistor level and evaluated delay by SPICE simulation (2004)



MWTAs (Minimum-Width Transistor Areas)

Balanced

→ Recent FPGA has 6-input LUTs, and more

Summary

- Conventional FPGA architecture
    - LUT
    - Channel
    - Hard Macro
        - DSP Block, BRAM, PLL, Processor...
- Programming Technology
    - SRAM, Anti-Fuse, Flash
- LUT Size Decision
    - Trade-off between area-performance

Exercise 3

1. (Mandatory) Investigate both Xilinx and Intel FPGA Architecture, what is new? and what is still remain?
2. (Mandatory) Why hard-macros are necessary on an FPGA ?
3. (Mandatory) Why PLLs are necessary on an FPGA?

Submit to OCW-i by PDF format

Deadline is 3rd, July, 2020 JST PM13:20

(At the beginning of the next lecture)