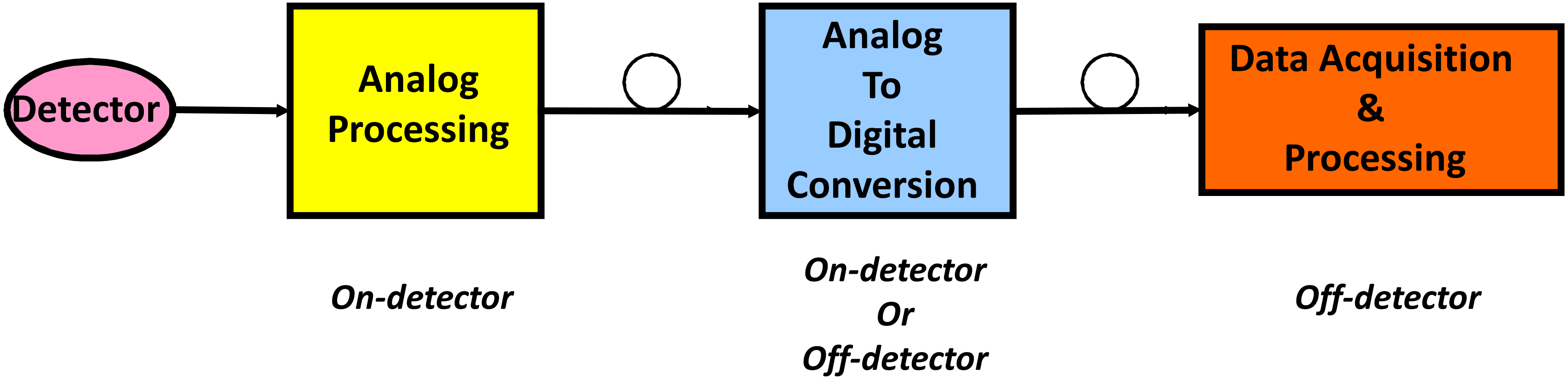
Introduction to Electronics in HEP Experiments

Philippe Farthouat

CERN

Outline

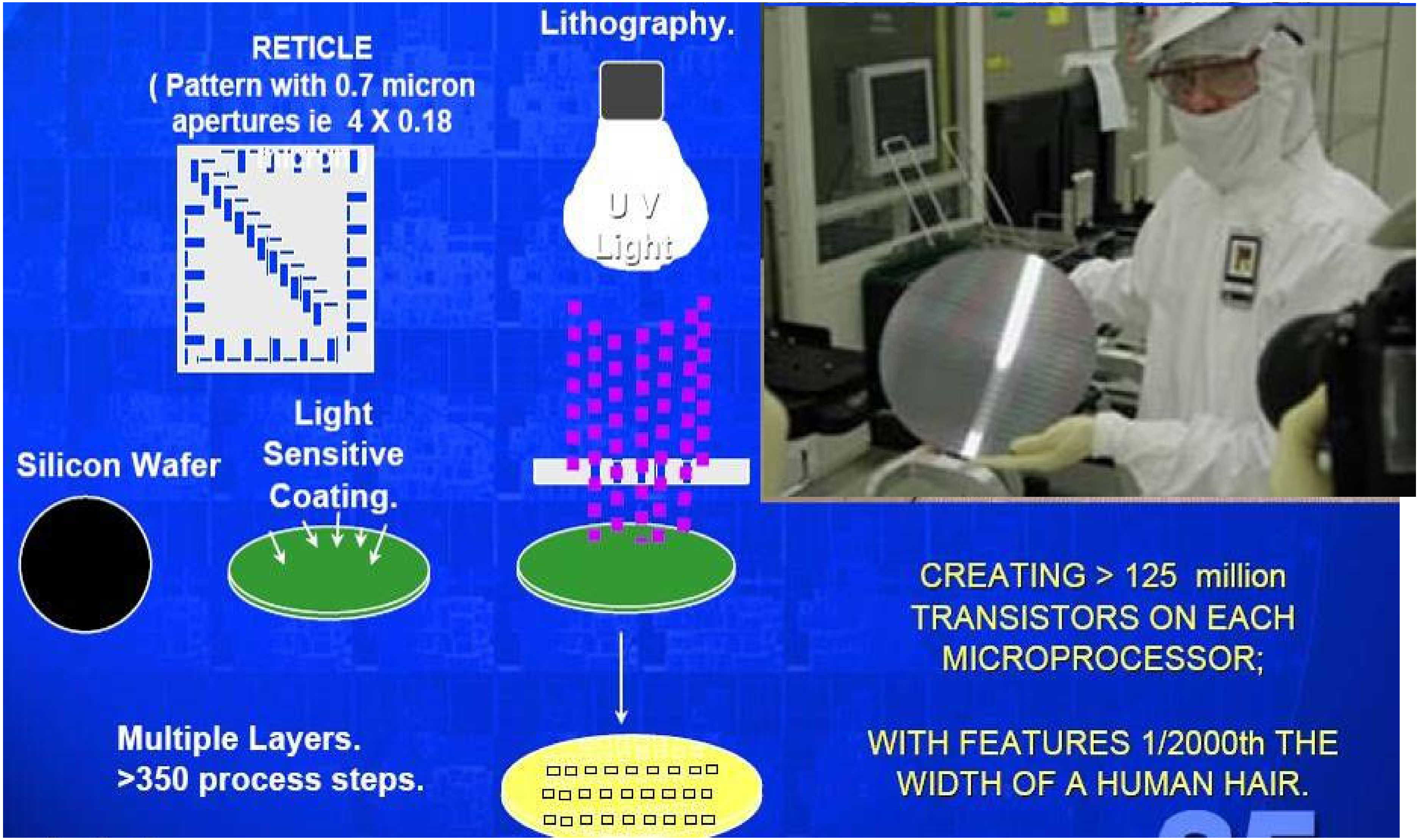

Detector
Analog Processing
Analog To Digital Conversion
Data Acquisition & Processing
On-detector
On-detector Or Off-detector
Off-detector

- Analog processing
- Analog to digital conversion
- Technology evolution
- Off-detector digital electronics

Processing of ASICs

© Intel

- From Sand to ICs…



Multiple Layers.
>350 process steps.

CREATING > 125 million TRANSISTORS ON EACH MICROPROCESSOR;

WITH FEATURES 1/2000th THE WIDTH OF A HUMAN HAIR.

Processing of ASICs

- For those interested in integrated circuits processing, please refer to F. Faccio presentation (from which I stole a few slides)

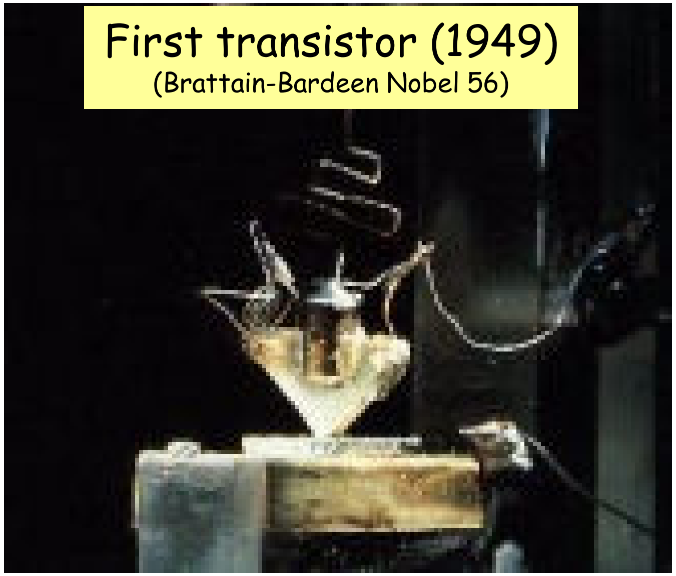
Step-by-step manufacturing of ULSI CMOS technologies

http://indico.cern.ch/conferenceDisplay.py?confId=48132

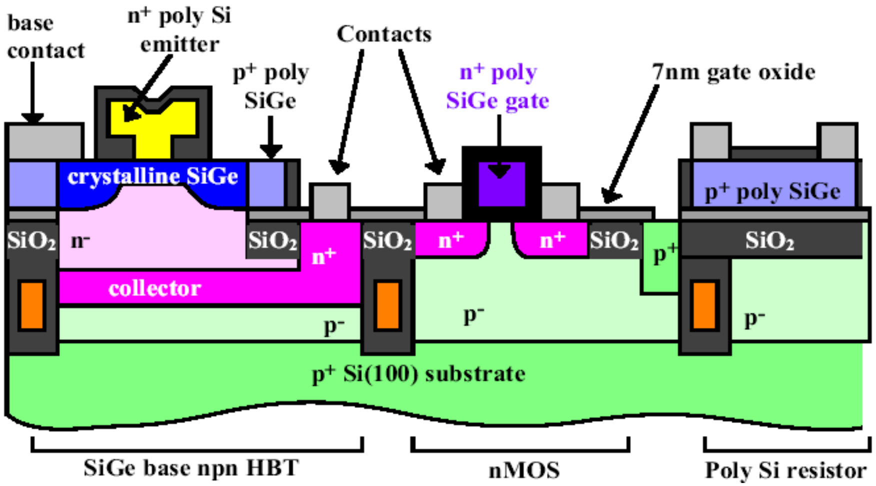
Evolution of technologies

First transistor (1949)

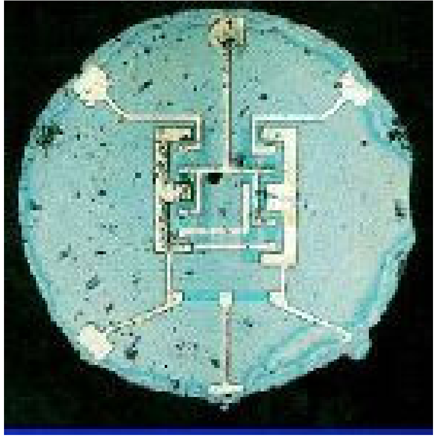
(Brattain-Bardeen Nobel 56)



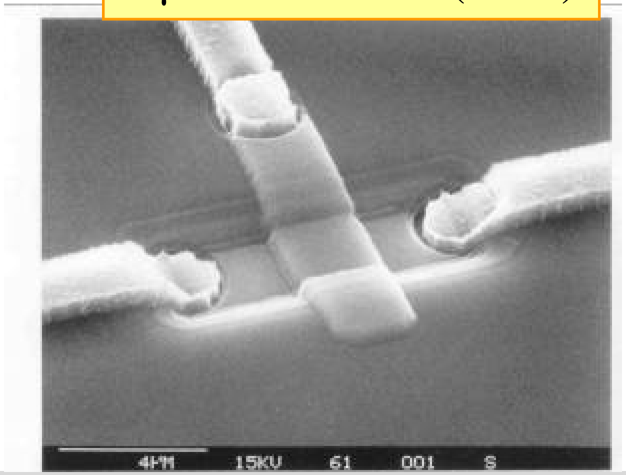
SiGe Bipolar in 0.35µm monolithic process



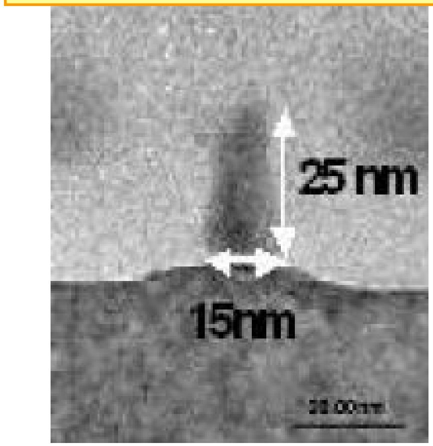
First planar IC (1961)



5 µm MOSFET (1985)

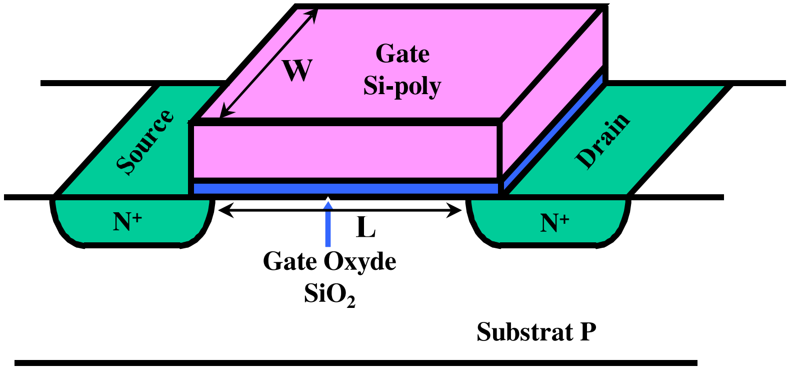


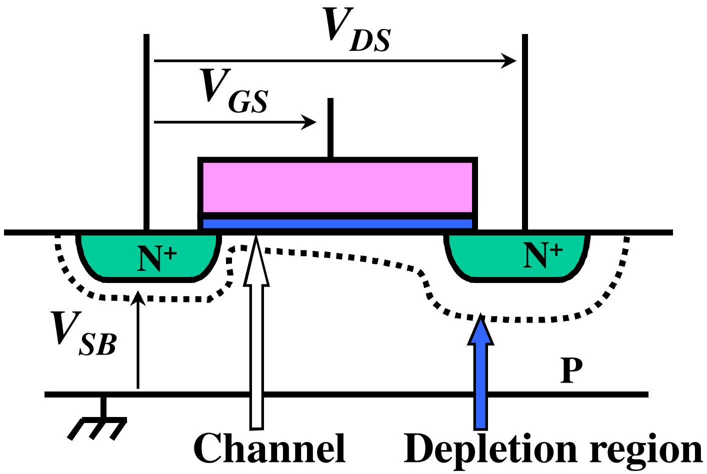
15 nm MOSFET (2005)



« CMOS scaling »

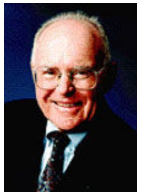
- Reduction of dimensions
  - Gate length : L
  - Oxide thickness : t_{ox}
- Improvement of speed in $1/L^2$
  - Transconductance : $g_m \alpha W/L$
  - Capacitance : $C \alpha WL$
  - speed : $F_T = g_m/C \alpha 1/L^2$
- Reduction of power dissipation
  - Capacitances decrease
  - Power supply level decreases
    - 2.5V in 0.25µm; 1.2V in 0.13µm
    - Current remains high!
- Reduction of costs (?)
  - Increase of integration density
  - Non recurrent engineering (NRE) costs increase





Principle of Nchannel MOSFET

Moore’s law



1965: Number of Integrated Circuit components will double every year

G. E. Moore, “Cramming More Components onto Integrated Circuits”, *Electronics*, vol. 38, no. 8, 1965.

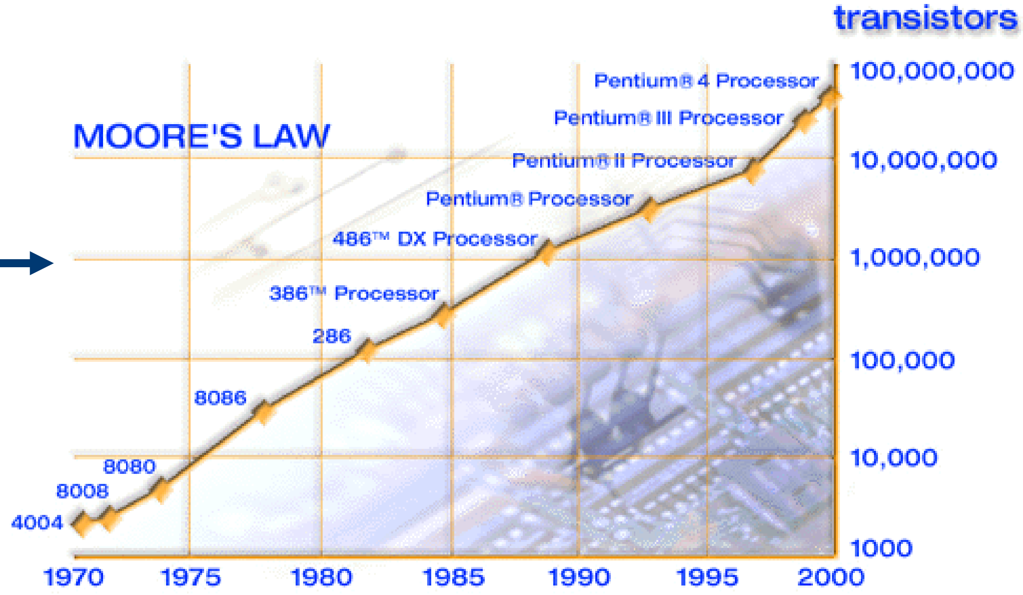
1975: Number of Integrated Circuit components will double every 18 months

G. E. Moore, “Progress in Digital Integrated Electronics”, *Technical Digest of the IEEE IEDM 1975*.

1996: The definition of “Moore’s Law” has come to refer to almost anything related to the semiconductor industry that when plotted on semi-log paper approximates a straight line. I don’t want to do anything to restrict this definition. - G. E. Moore, 8/7/1996

P. K. Bondyopadhyay, “Moore’s Law Governs the Silicon Revolution”, *Proc. of the IEEE*, vol. 86, no. 1, Jan. 1998, pp. 78-81.

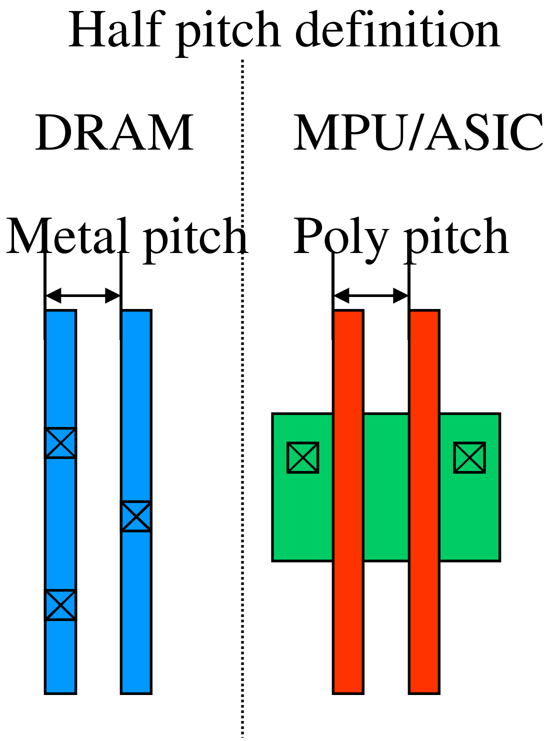
An example:
Intel’s Microprocessors



http://www.intel.com/

Moore’s law fundamentals

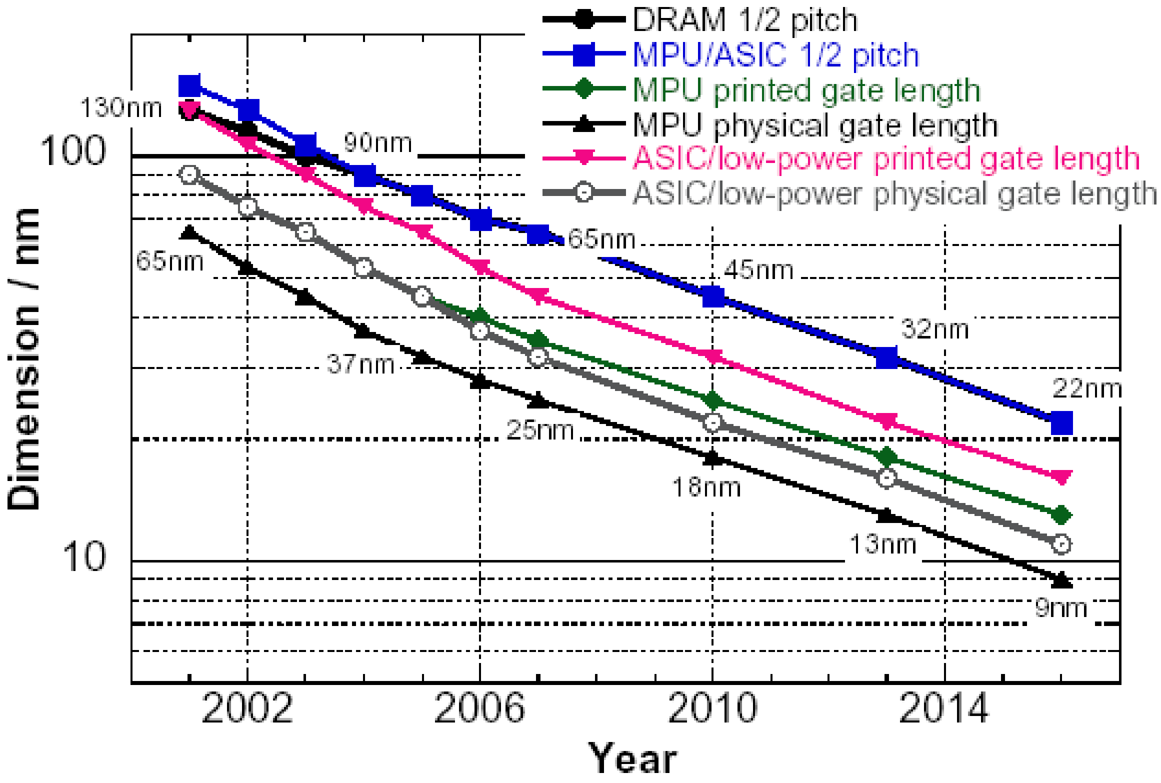
- For every generation:
  - Characteristic Dimensions x 0.7
  - Area x 0.5
  - Chip size x 1.5
  - Structural improvement x 1.3
  - N of components x 4
  - Clock frequency x 1.4



Technology nodes (1/2 pitch):

$$250 \rightarrow 180 \rightarrow \underbrace{\overbrace{130 \rightarrow 90}^{0.7}\ \overbrace{\rightarrow 65}^{0.7}}_{0.5} \rightarrow 45 \rightarrow 32 \rightarrow 22 \rightarrow 16$$

CMOS technology scaling



This roadmap is 7 years old: Now 45 nm is in production

CMOS technology

- Details can be found in the following presentations

IEDM summary and outlook (Walter Snoeys)

http://indico.cern.ch/conferenceDisplay.py?confId=49285

Summary from ISSCC09 *(Alessandro Marchioro)*

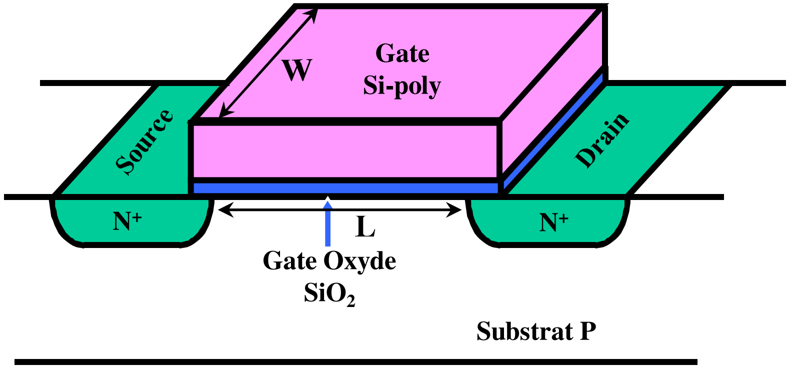
http://indico.cern.ch/conferenceDisplay.py?confId=48130

Impact for our applications

- What do we gain?
  - Radiation hardness
  - Integration
  - Power dissipation
  - Speed
- What do we loose?
  - Non recurrent engineering (NRE) costs are high
    - Order of 0.5MChF for 0.13µm CMOS technology
    - Wafer cost is low but our production volume is low
    - Limit as much as possible the iterations (prototyping)
    - Limit the number of different designs
  - Design tools are extremely complex
    - Long learning curve
    - High investment

Gain on radiation hardness

- Radiation hardness is an issue for a number of our applications
  - e.g. LHC experiments tracker electronics can receive as much as 100Mrad (1MGy) during their lifetime
- In CMOS technology most of the problems are coming from charge trapping in the gate oxyde
  - When the gate is very thin, there is less or no problems

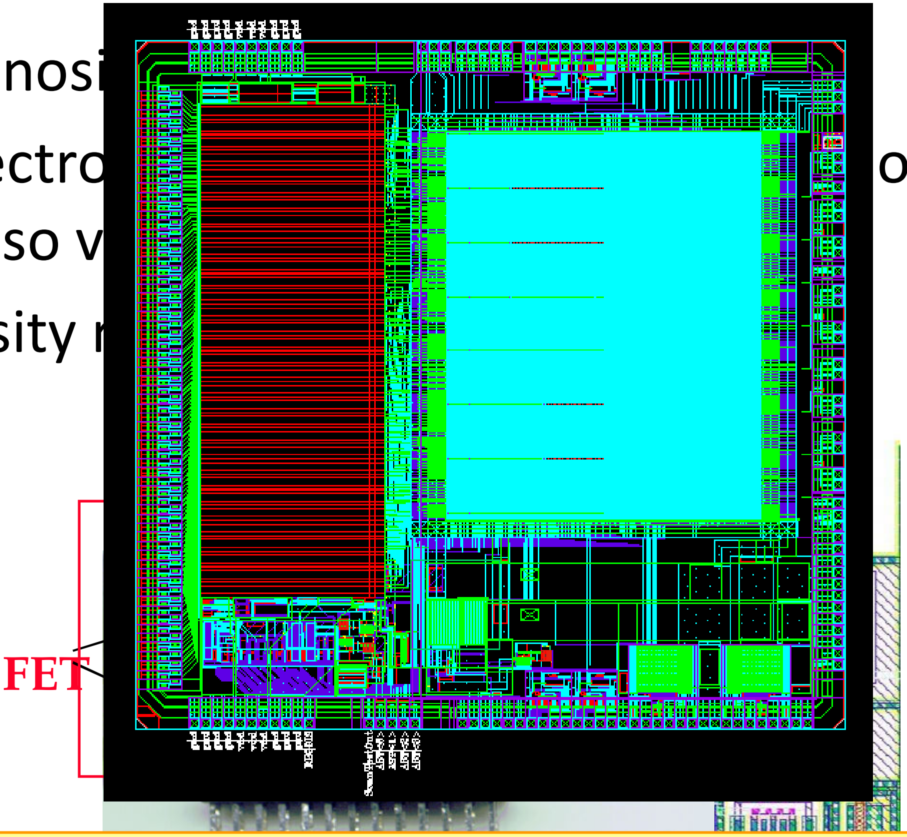


- Those interested in the subject can look at a presentation from F. Faccio

  http://indico.cern.ch/materialDisplay.py?contribId=34&materialId=slides&confId=43007

Gain in integration (1)

- Detectors have more and more channels
  - High luminosi
- Space for electro owed quantity of material is also v
  - High density r



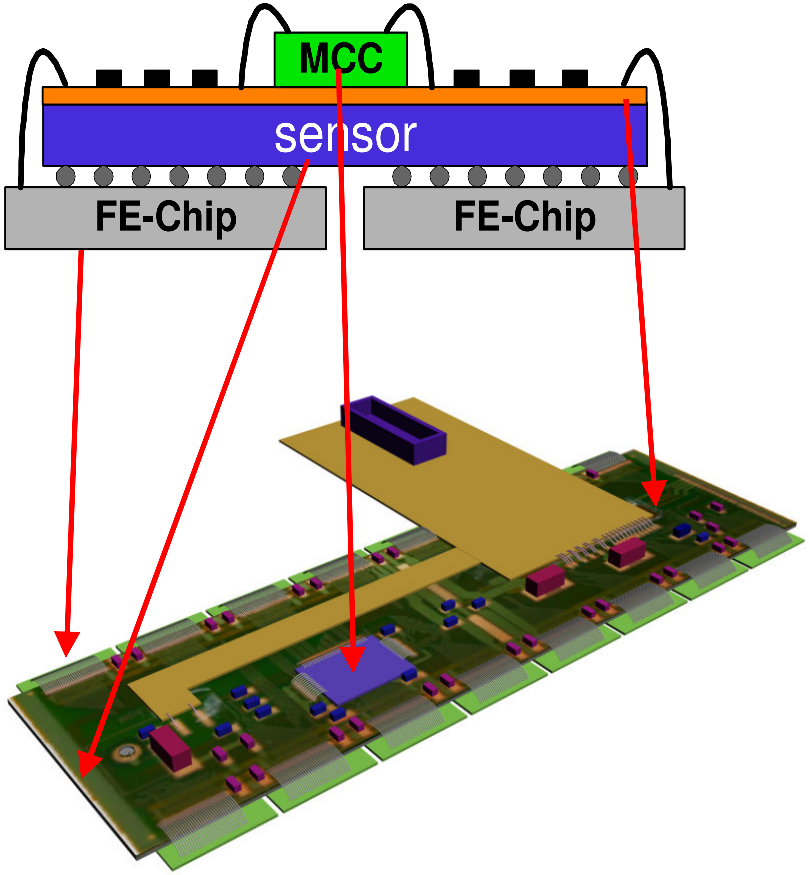
128-channel charge preamp, shaper, discriminator, pipeline and data compression
0.25µm CMOS technology
7.5 * 7.2 mm^2

← 100 µm →

Charge preamp in 0.8µm BiCMOS

Gain in integration (2)

- Pixel detectors require the readout electronics bump bond on the sensor
- Space available for the readout electronics is equal to the size of the pixel
  - ATLAS case 400*50 µm^2 today
  - Smaller for upgrade 250*50 µm^2
- Smaller feature size technology needed
  - Current readout chip in 0.25µm CMOS technology
  - Next in 0.13µm



Sketch of an ATLAS pixel module

Gain in power

- Power is always a problem

  (see for instance Power distribution in future experiments (F. Faccio) http://indico.cern.ch/conferenceDisplay.py?confId=39721)

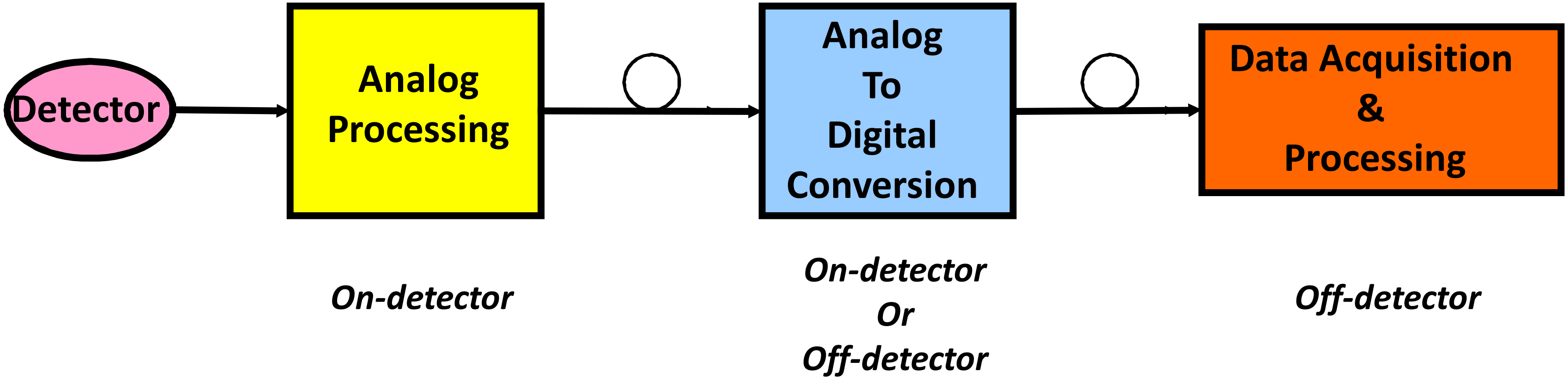
  - Heat dissipation in enclosed volume
  - Cables needed to bring the power in
  - Even 1mW per channel for a tracker leads to 60kW in an “LHC like” tracker
- The wish of having the ADC as early as possible in the readout chain requires low power ADC
- Standard metric to characterize an ADC is the “Figure of Merit”

$$FOM = \frac{Power}{2^{ENB} \times F_{sampling}}$$

	0.25µm FOM=1pJ/conv	65nm FOM=50fJ/conv	Saving in Power Supply (8ChF/W)
ATLAS Calorimeter 200k channels 15-ENB 40MHz	262 KW	13 KW	~2 MChF
Linear Collider Calorimeter 15M channels 13-ENB 50MHz	6.4 MW	340 kW	A lot

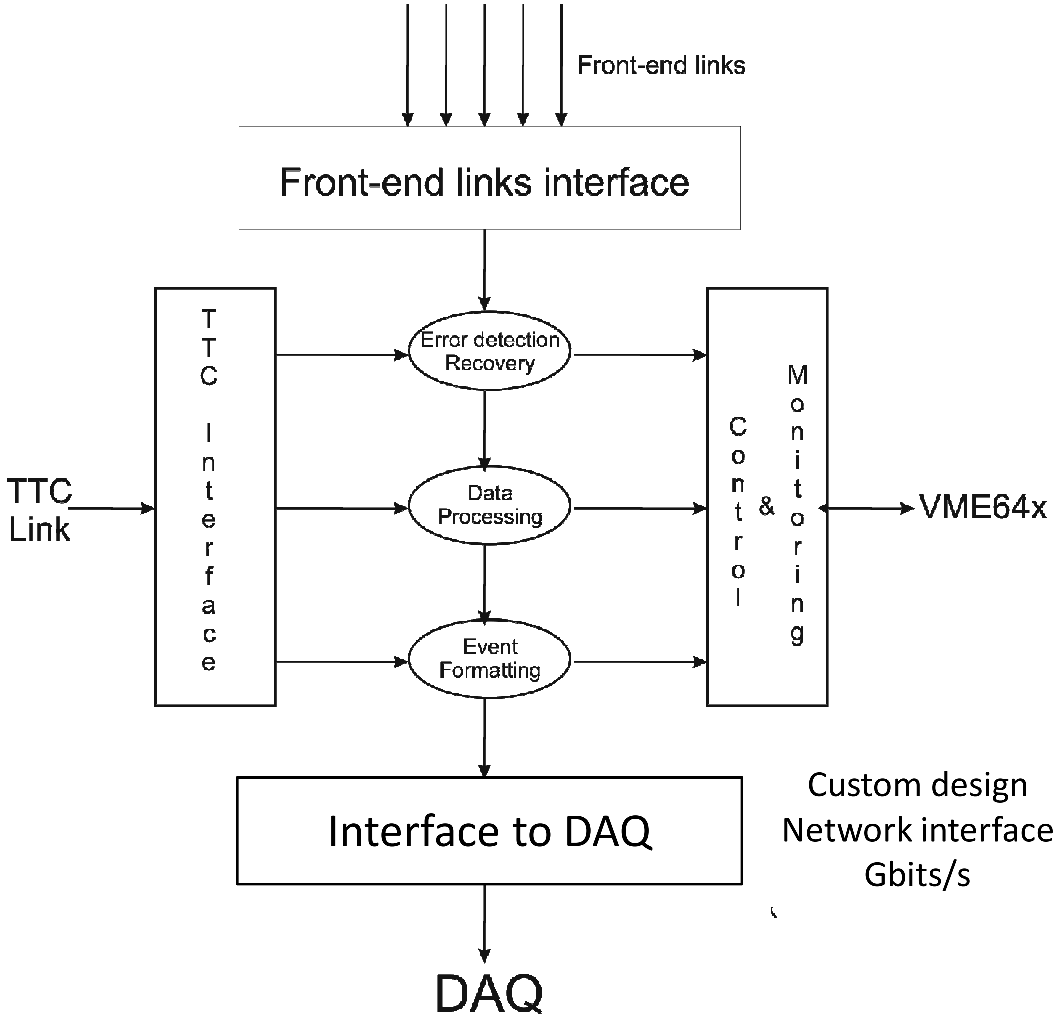
Courtesy A. Marchioro CERN

Outline


Detector
Analog Processing
Analog To Digital Conversion
Data Acquisition & Processing
On-detector
On-detector Or Off-detector
Off-detector

- Analog processing
- Analog to digital conversion
- Technology evolution
- Off-detector digital electronics

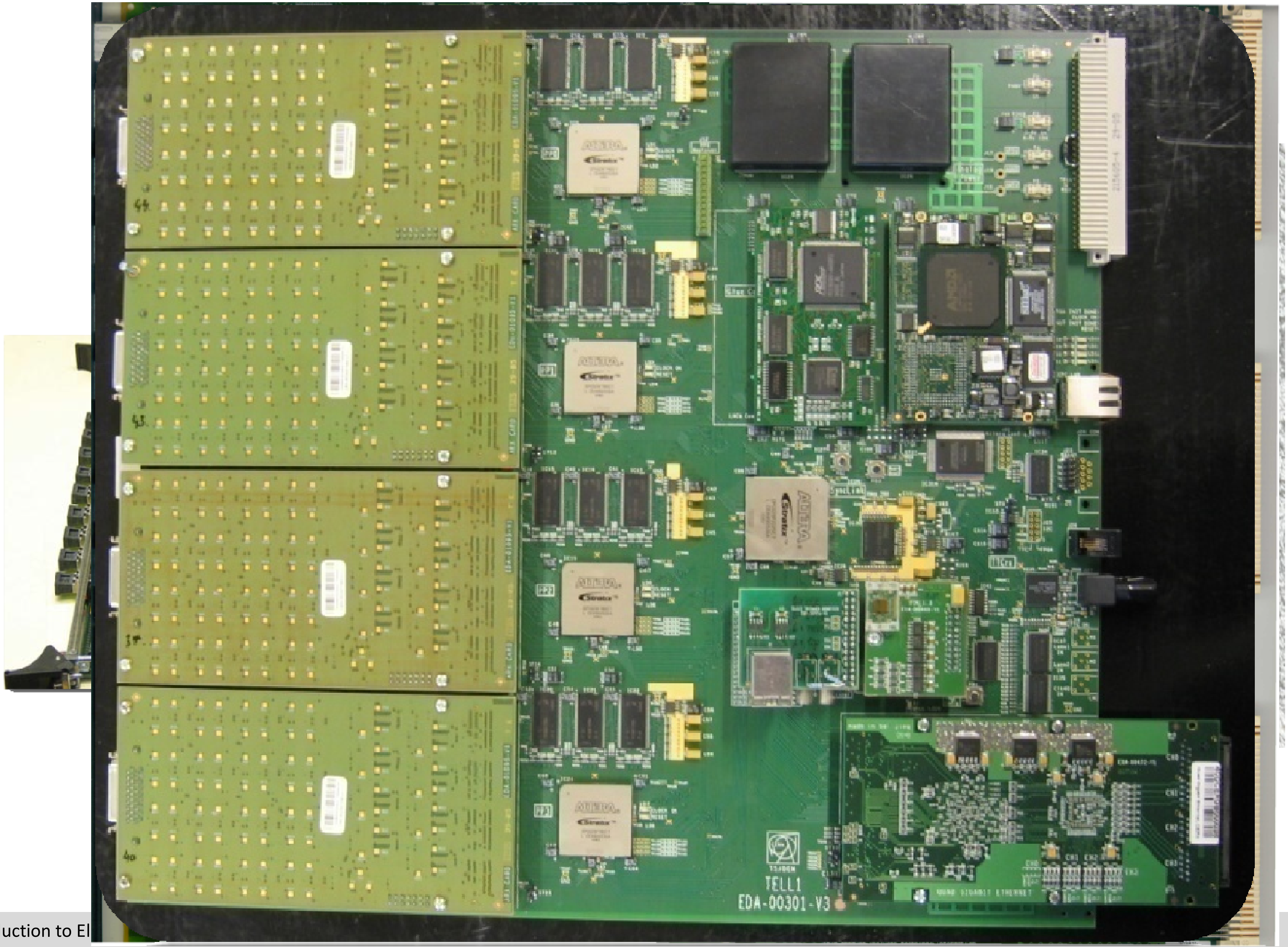
Off-detector readout electronics



Custom design
Network interface
Gbits/s

- Generic block diagram

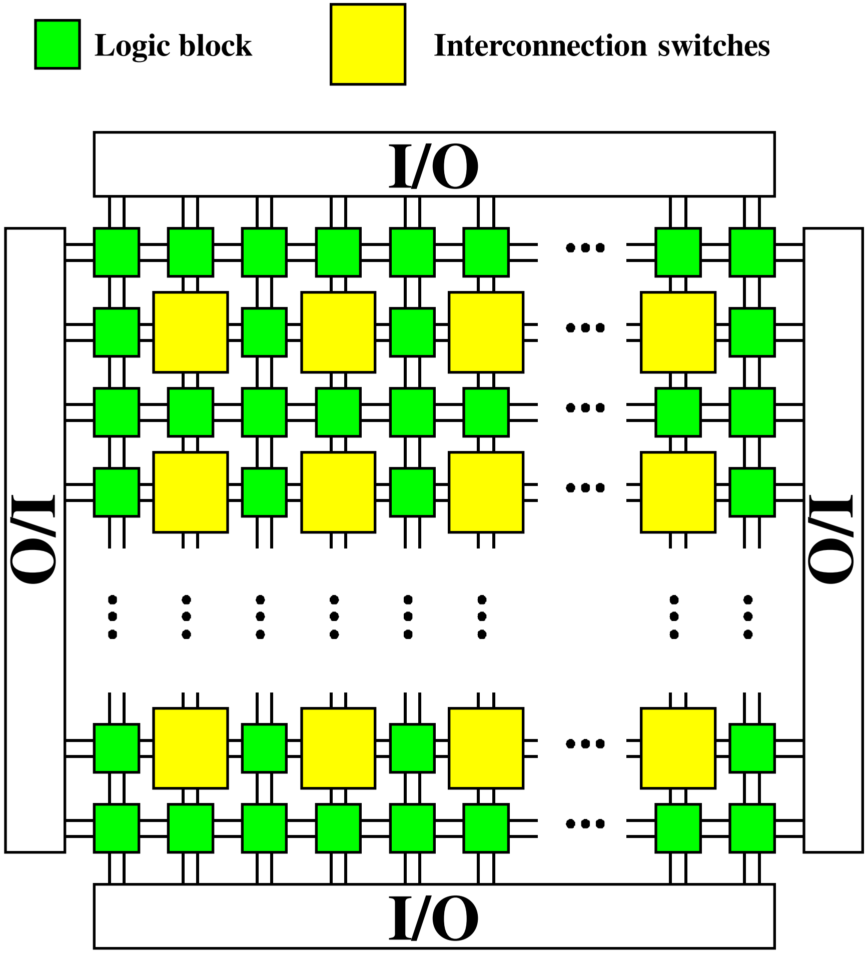
Examples of modules



FPGA - Generic Structure

FPGA building blocks:

- Programmable logic blocks
  Implement combinatorial and sequential logic
- Programmable interconnect
  Wires to connect inputs and outputs to logic blocks
- Programmable I/O blocks
  Special logic blocks at the periphery of device for external connections

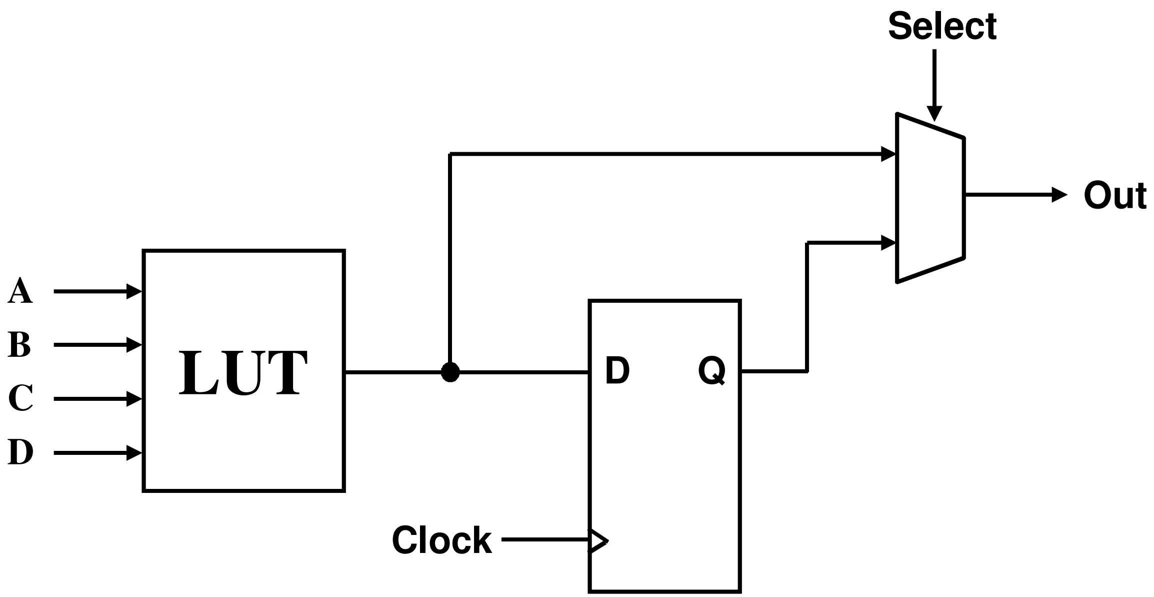


Other FPGA Building Blocks

- Clock distribution
- Embedded memory blocks
- Special purpose blocks:
  - DSP blocks:
    - Hardware multipliers, adders and registers
  - Embedded microprocessors/microcontrollers
  - High-speed serial transceivers

FPGA – Basic Logic Element

- LUT to implement combinatorial logic
- Register for sequential circuits
- Additional logic (not shown):
  - Carry logic for arithmetic functions
  - Expansion logic for functions requiring more than 4 inputs

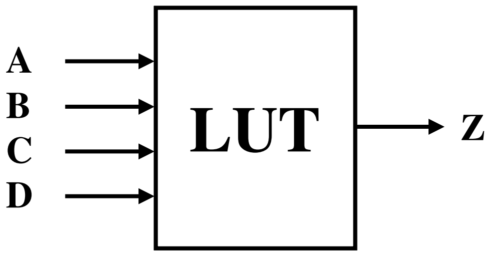


Look-Up Tables (LUT)

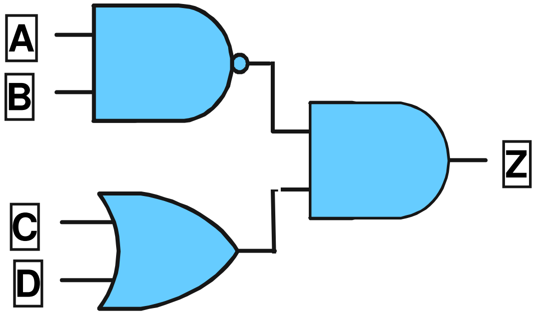
- Look-up table with N-inputs can be used to implement any combinatorial function of N inputs
- LUT is programmed with the truth-table

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0

Truth-table



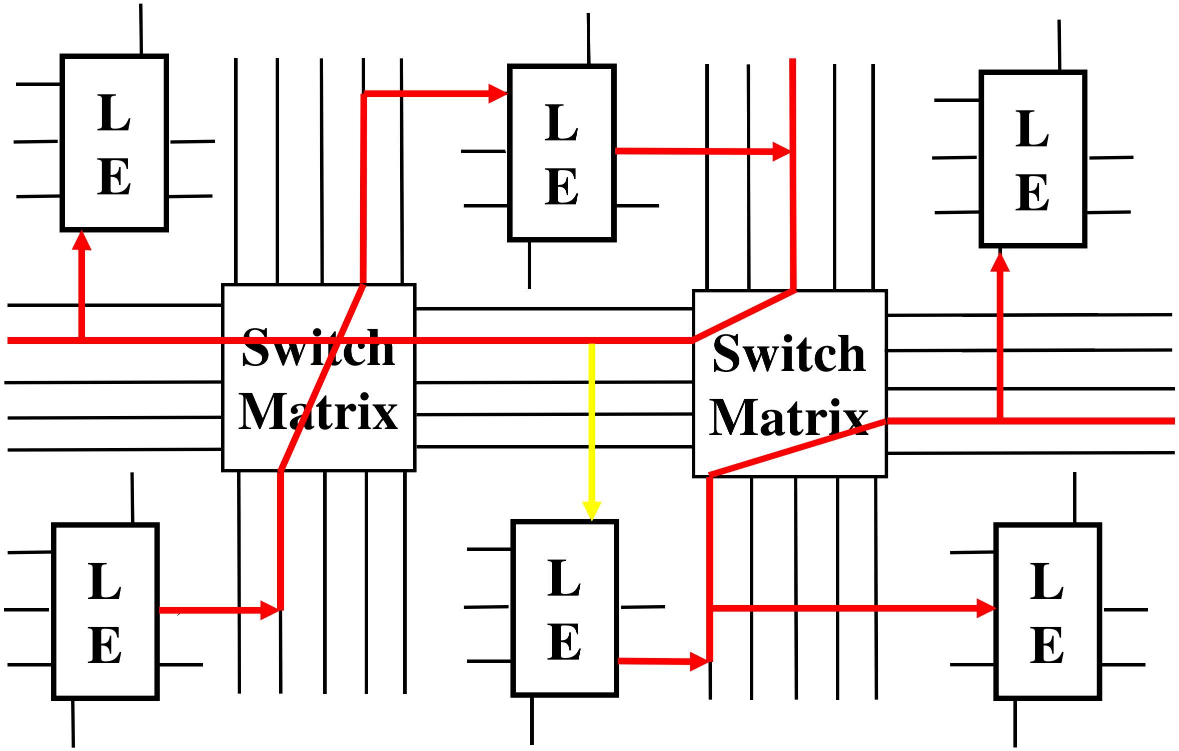
LUT implementation



Gate implementation

Programmable Interconnect

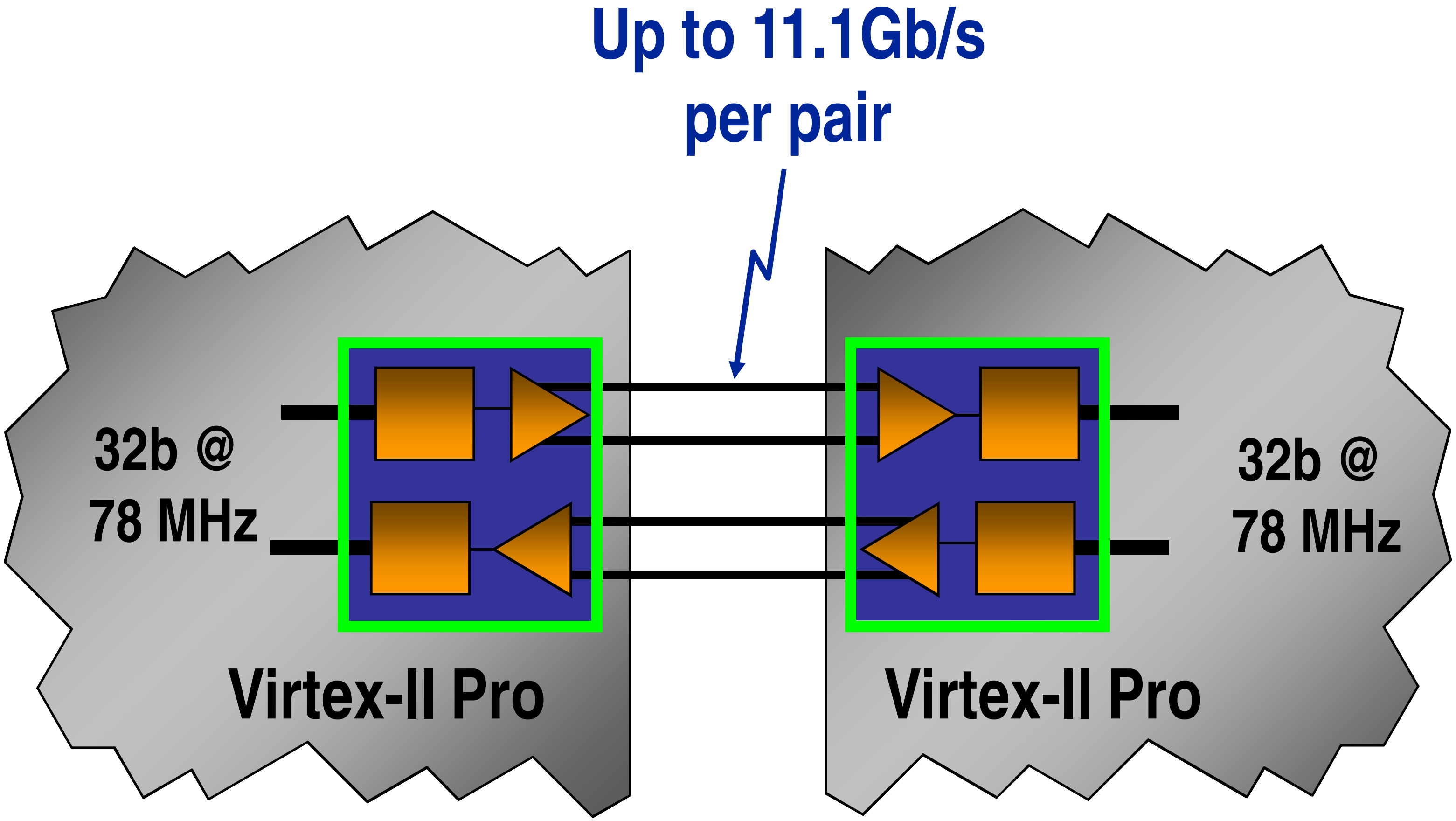
- Interconnect hierarchy (not shown)
  - Fast local interconnect
  - Horizontal and vertical lines of various lengths



Configuration Storage Elements

- Static Random Access Memory (SRAM)
  - each switch is a pass transistor controlled by the state of an SRAM bit
  - FPGA needs to be configured at power-on
- Flash Erasable Programmable ROM (Flash)
  - each switch is a floating-gate transistor that can be turned off by injecting charge onto its gate. FPGA itself holds the program
  - reprogrammable, even in-circuit
- Fusible Links (“Antifuse”)
  - Forms a low resistance path when electrically programmed
  - one-time programmable in special programming machine
  - radiation tolerant

Xilinx: Rocket I/O



FPGA Vendors & Device Families

- Xilinx
  - Virtex-II/Virtex-4: Feature-packed high-performance SRAM-based FPGA
  - Spartan 3: low-cost feature reduced version
  - CoolRunner: CPLDs
- Altera
  - Stratix/Stratix-II
    - High-performance SRAM-based FPGAs
  - Cyclone/Cyclone-II
    - Low-cost feature reduced version for cost-critical applications
  - MAX3000/7000 CPLDs
  - MAX-II: Flash-based FPGA
- Actel
  - Anti-fuse based FPGAs
    - Radiation tolerant
  - Flash-based FPGAs
- Lattice
  - Flash-based FPGAs
  - CPLDs (EEPROM)
- QuickLogic
  - ViaLink-based FPGAs

Latest Devices: Capacity & Features

Xilinx Virtex-6

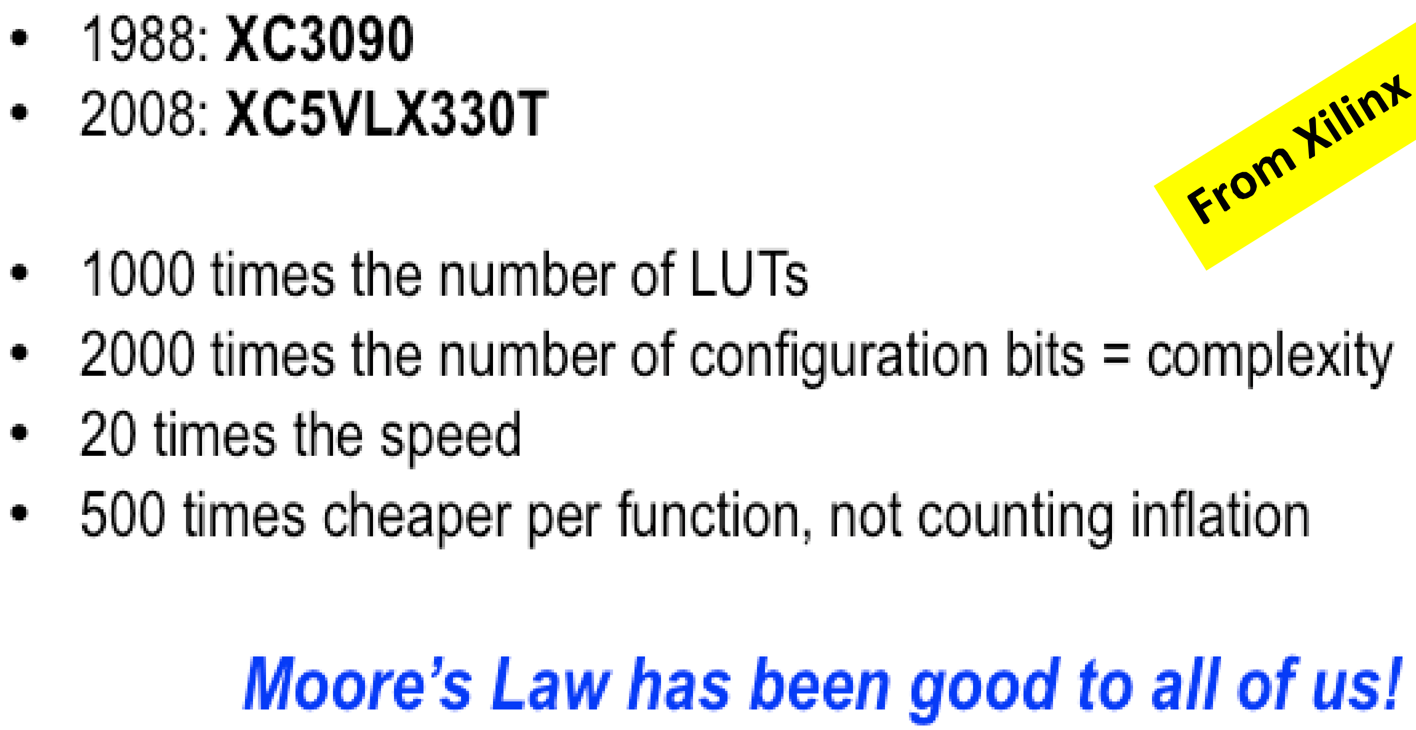
- 40nm process
- Up to 1200 I/Os
- Up to 760k logic cells
- Up to 38Mb embedded RAM
- Up to 2000 DSP slices
- Up to 36 high-speed serial transceivers at 11.1GB/s

Altera Stratix-IV

- 40nm process
- Up to 1104 I/Os
- Up to 680k logic elements
- Up to 22.4Mb embedded RAM
- Up to 1288 18x18 multipliers
- Up to 48 Serial I/O at 11.5Gb/s

Plus processors (PowerPC)

FPGAs evolution



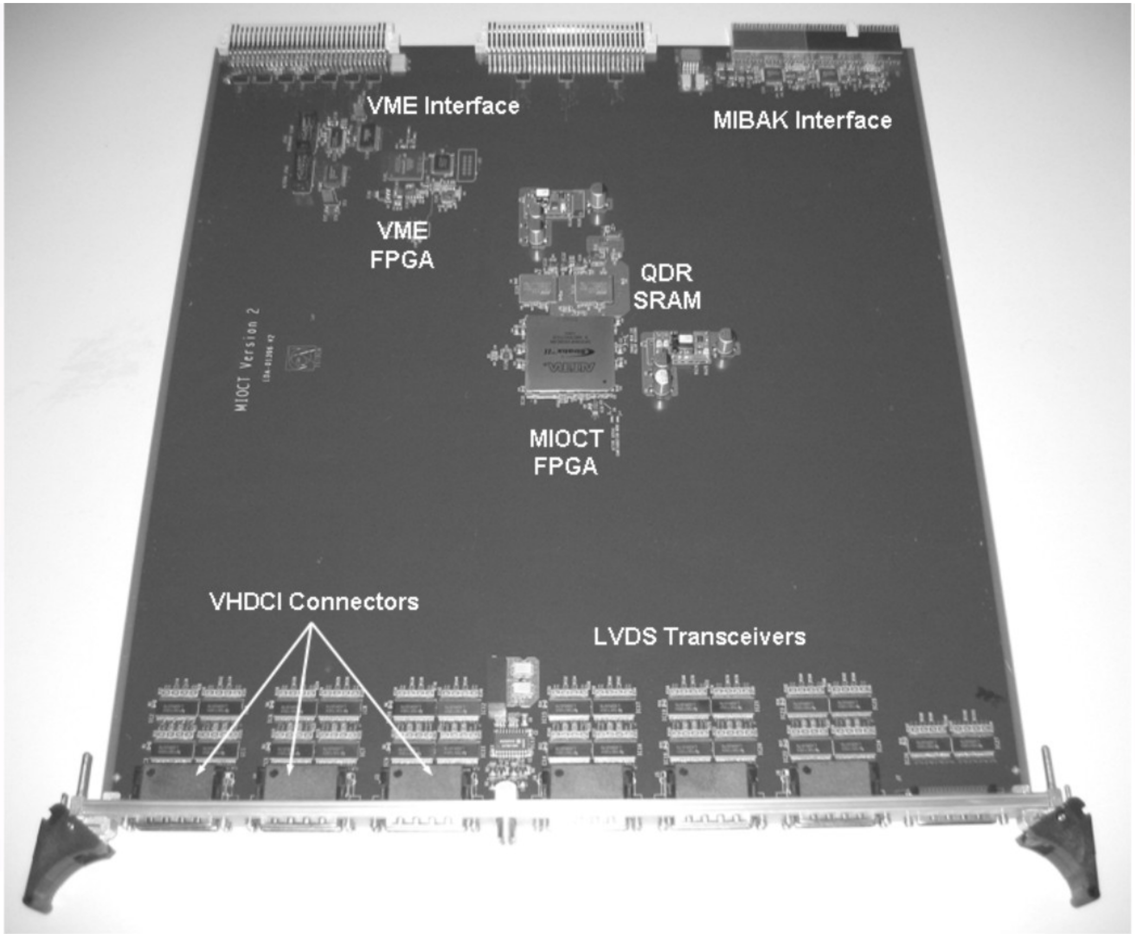
- 1988: **XC3090**
- 2008: **XC5VLX330T**

From Xilinx

- 1000 times the number of LUTs
- 2000 times the number of configuration bits = complexity
- 20 times the speed
- 500 times cheaper per function, not counting inflation

Moore's Law has been good to all of us!

Example of evolution



Demonstrator at the end of the 90's

Final version installed

- Muon trigger board for ATLAS
  - Handles 13 input links, each of them receiving 32-bit every 25ns
  - ~17 Gb/s processed

VHDL Language

- Hardware Description Language (HDL)
  - High-level language for to model, simulate, and synthesize digital circuits and systems.
- History
  - 1980: US Department of Defense Very High Speed Integrated Circuit program (VHSIC)
  - 1987: Institute of Electrical and Electronics Engineers ratifies IEEE Standard 1076 (VHDL'87)
  - 1993: VHDL language was revised and updated
- Verilog is the other major HDL
  - Syntax similar to C language
- At CERN VHDL is mostly used for FPGA design
- Many tools accept both Verilog and VHDL

Terminology

- Behavioral modeling
  - Describes the functionality of a component/system
  - For the purpose of simulation and synthesis
- Structural modeling
  - A component is described by the interconnection of lower level components/primitives
  - For the purpose of synthesis and simulation
- Synthesis:
  - Translating the HDL code into a circuit, which is then optimized
- Register Transfer Level (RTL):
  - Type of behavioral model used for instance for synthesis

Digital Circuits and VHDL Primitives

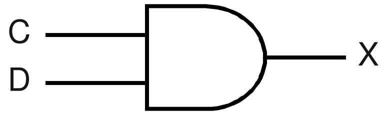
- Most digital systems can be described based on a few basic circuit elements:
  - Combinational Logic Gates:
    - NOT, OR, AND
  - Flip Flop
  - Latch
  - Tri-state Buffer

- Each circuit primitive can be described in VHDL and used as the basis for describing more complex circuits.

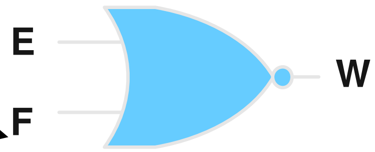
Digital Circuit Primitives

- Combinational Logic Gates: NOT, OR, AND
- Flip Flop/Latch
- Tri-state Buffer
- Logic gates can be modeled using concurrent signal assignments:

```
Z <= not A;
Y <= A or B;
X <= C and D;
W <= E nor F;
U <= B nand D;
V <= C xor F;
```



AND



NOR

- It is possible to design circuits from logic gates in this way
- For design entry it is preferable to use other VHDL structures that allow circuit descriptions at a higher level of abstraction

Combinatorial Logic: Decoder

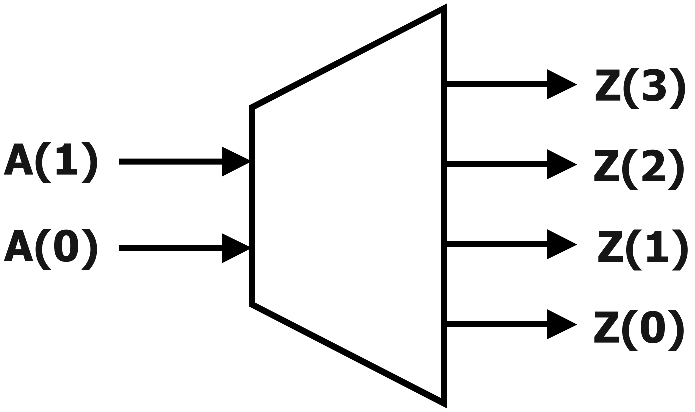
Example: 2-to-4 decoder

```
entity decoder is
  port (
    A : in  std_logic_vector(1 downto
   0);
    Z : out std_logic_vector(3 downto
   0)
  );
end entity decoder;



architecture when_else of decoder is
begin
  Z <= "0001" when A = "00" else
       "0010" when A = "01" else
       "0100" when A = "10" else
       "1000" when A = "11" else
       "XXXX";
end architecture when_else;
```

Interface

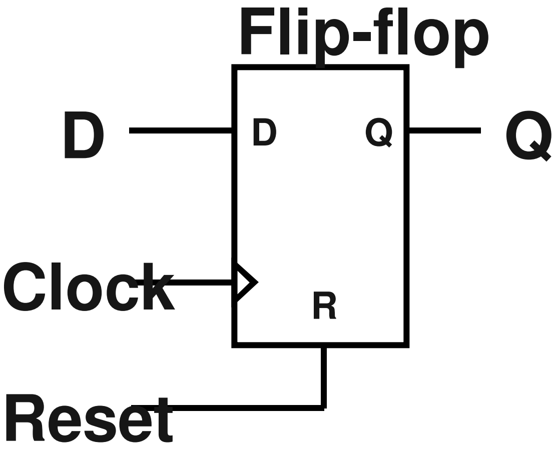


Functionality

A(1..0)		Z(3..0)			
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

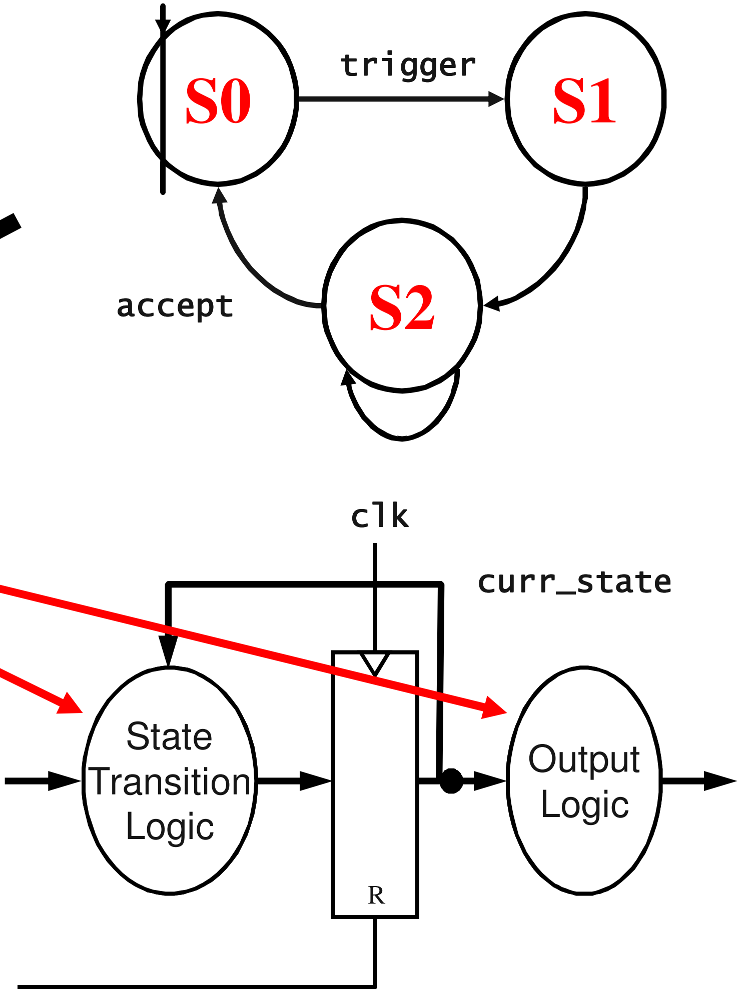
Sequential Logic: D-Flip Flop

```
architecture rtl of D_FF is
begin
  process (Clock, Reset) is
  begin
    if Reset = ‘1’ then
      Q <= ‘0’;
    if rising_edge(Clock)
   then
      Q <= D;
    end if;
  end process;
end architecture rtl;
```

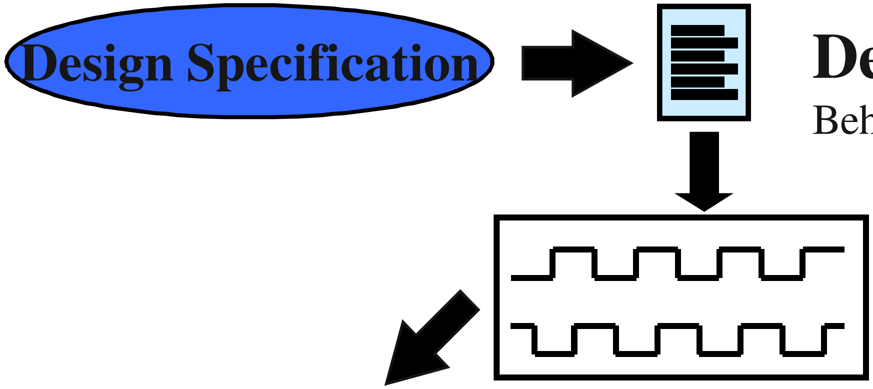


State Machine (cont.)

```
process (cur_state, trigger, accept)
   is
begin
  case cur_state is
    when s0 =>
      active <= '0';
      if (trigger = '1') then
        next_state <= s1;
      else
        next_state <= s0;
      end if;
    when s1 =>
      active <= '1';
      next_state <= s2;
    when s2 =>
      active <= '1';
      if (accept = '1') then
        next_state <= s0;
      else
        next_state <= s2;
      end if;
  end case;
end process;
```



FPGA Design Flow



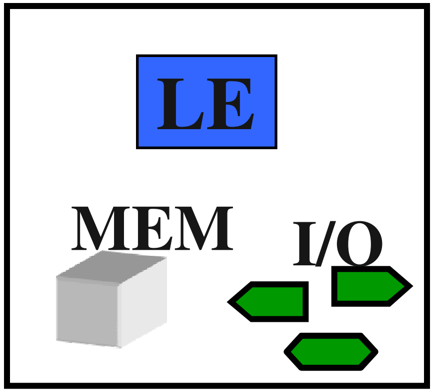
Design Entry/RTL Coding

Behavioral or Structural Description of Design

RTL Simulation

- Functional Simulation
- Verify Logic Model & Data Flow

(No Timing Delays)



Synthesis

- Translate Design into Device Specific Primitives
- Optimization to Meet Required Area & Performance Constraints

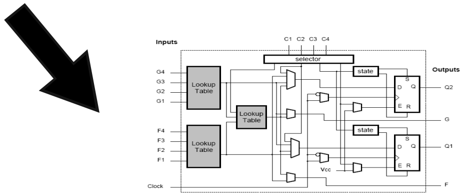
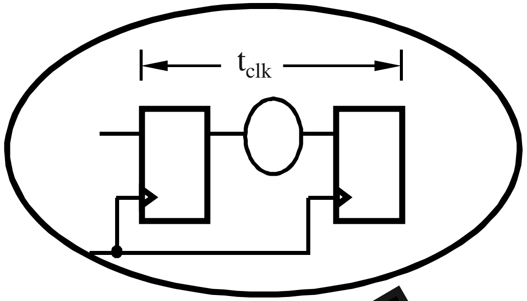


Figure 18 - *Xilinx XC4000 Configurable Logic Block (CLB).*

Place & Route

- Map Primitives to Specific Locations inside
  Target Technology with Reference to Area &
- Performance Constraints
- Specify Routing Resources to Be Used

FPGA Design Flow

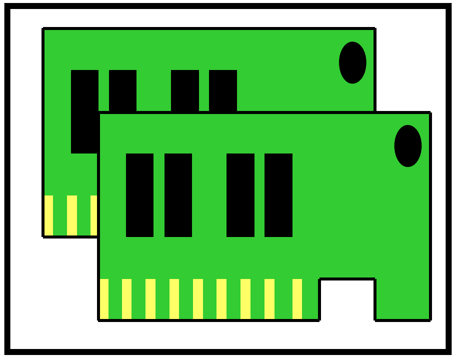


Timing Analysis

- Verify Performance Specifications Were Met
- Static Timing Analysis

Gate Level Simulation

- Timing Simulation
- Verify Design Will Work in Target Technology

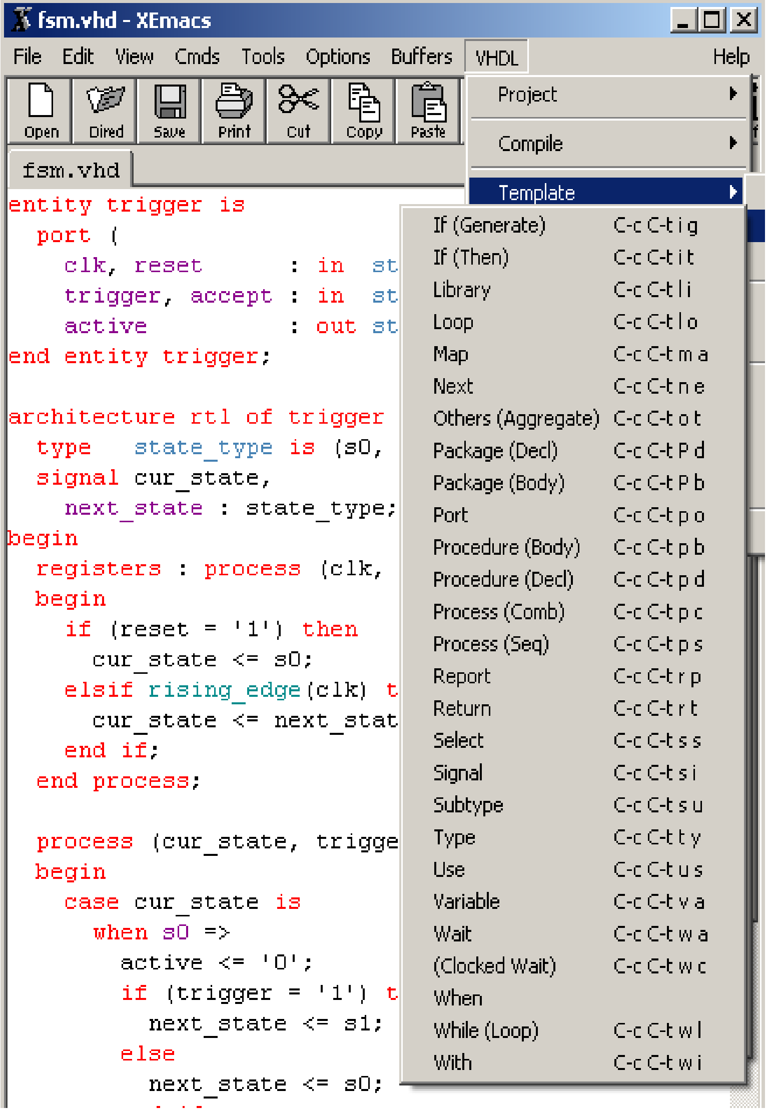


Program & Test

- Program & Test Device on Board

Text-based: emacs VHDL-mode

- Special mode for editing VHDL source files in emacs
- Features:
  - Syntax colouring
  - Automatic completions
  - Automatic indentation
  - Templates for all VHDL constructs
  - Launching external VHDL compiler



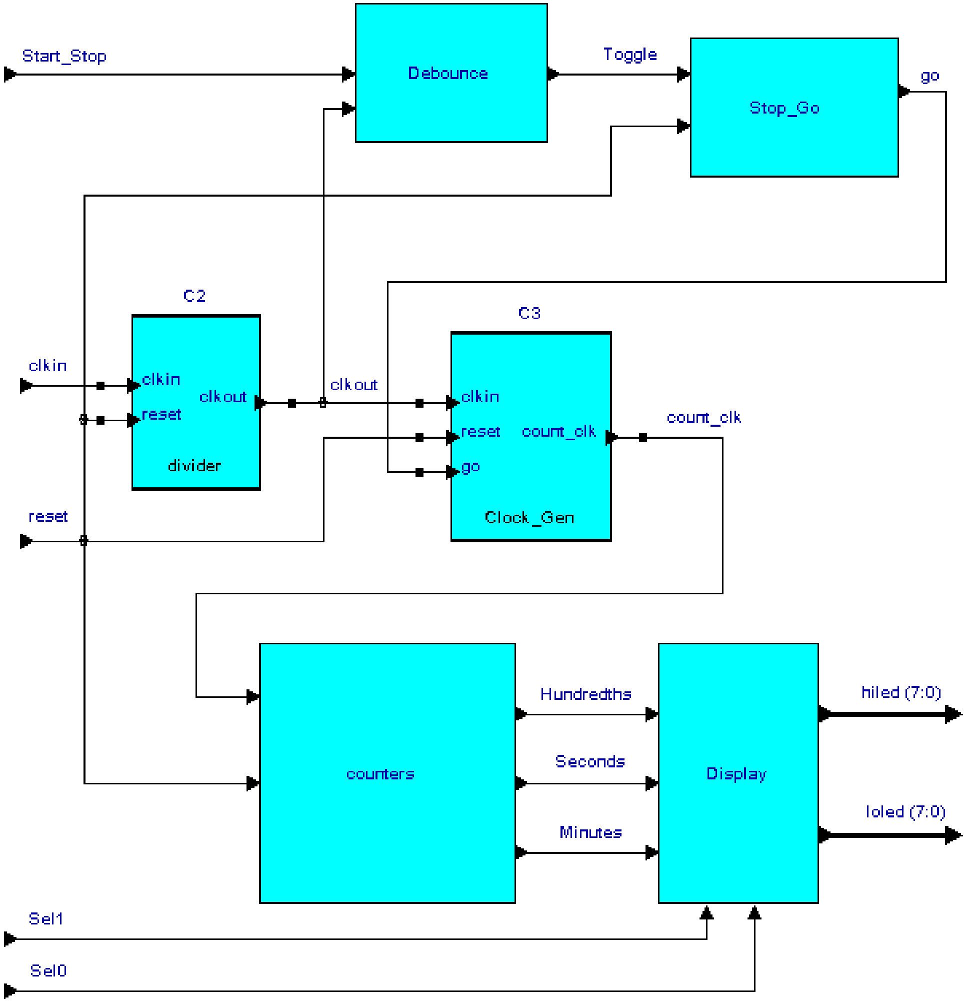
```
entity trigger is
  port (
    clk, reset        : in  st
    trigger, accept : in  st
    active            : out st
end entity trigger;

architecture rtl of trigger
  type   state_type is (s0,
  signal cur_state,
    next_state : state_type;
begin
  registers : process (clk,
  begin
    if (reset = '1') then
      cur_state <= s0;
    elsif rising_edge(clk) t
      cur_state <= next_stat
    end if;
  end process;

  process (cur_state, trigge
  begin
    case cur_state is
      when s0 =>
        active <= '0';
        if (trigger = '1') t
          next_state <= s1;
        else
          next_state <= s0;
```

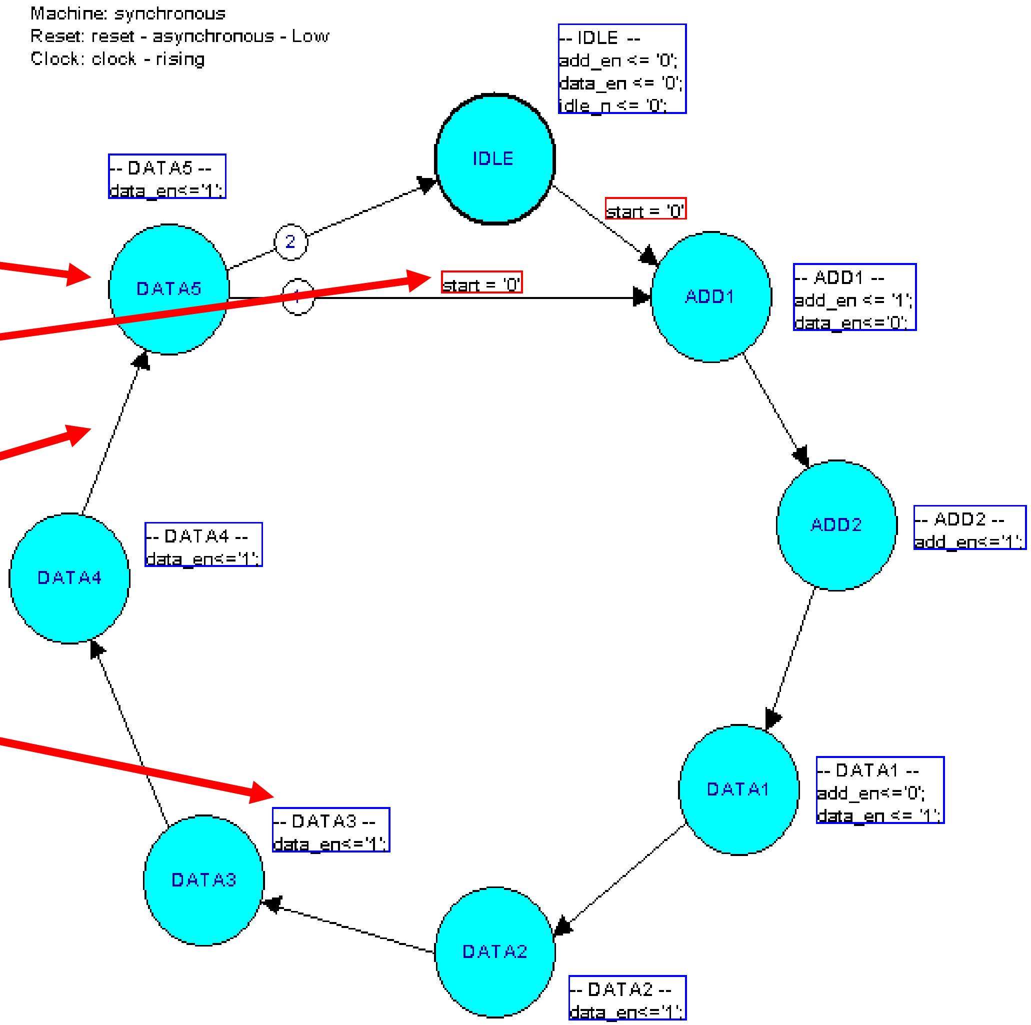
Block Diagram

- Hierarchical design methoc
  - top-down
  - bottom-up
- Contents of a block can be type of design unit
- Top-level block diagram:
  - Partitioning of the design
  - Connections between the underlying HDL design units



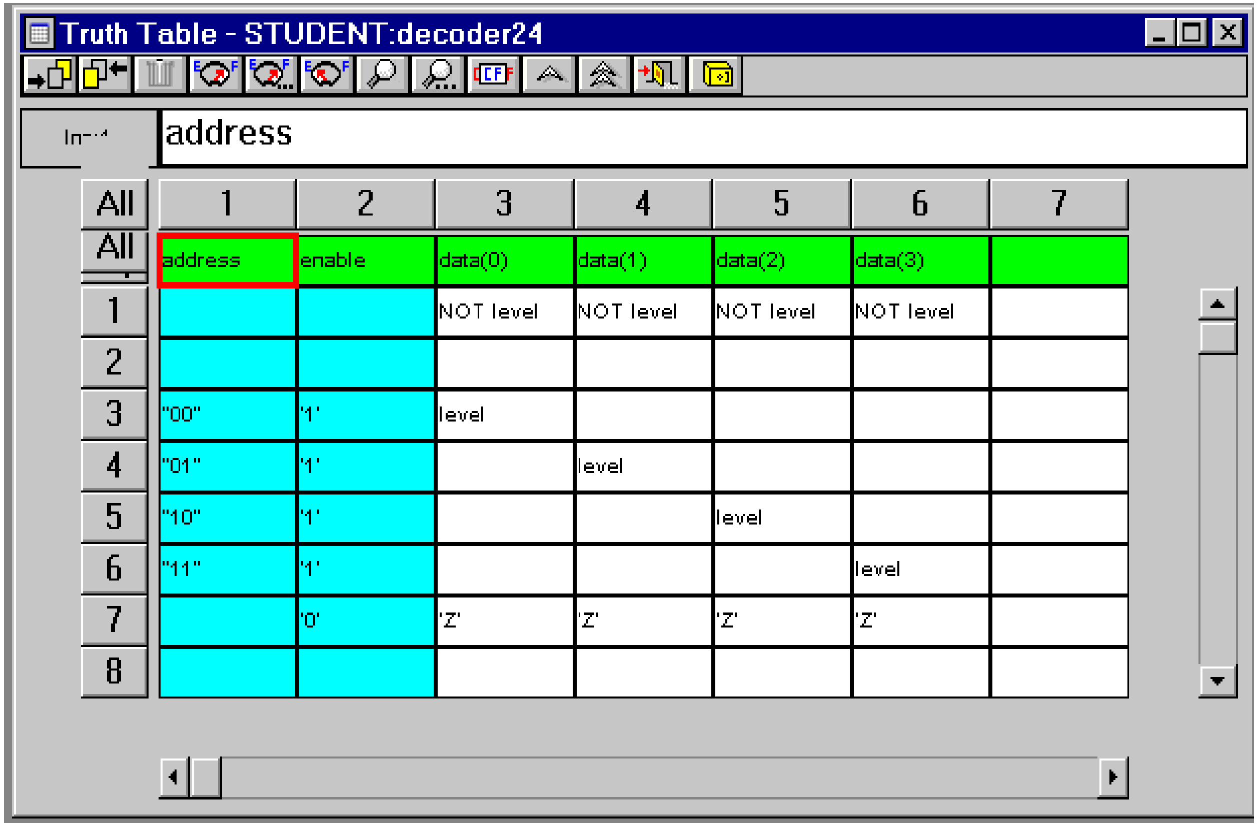
State Diagram

- "Bubble" diagram
  - States
  - Conditions
  - Transitions
  - Outputs
- Useful for developing control modules



Truth Table

- Normally used to describe combinatorial logic
- Can also be used for sequential circuits (e.g. state machines)



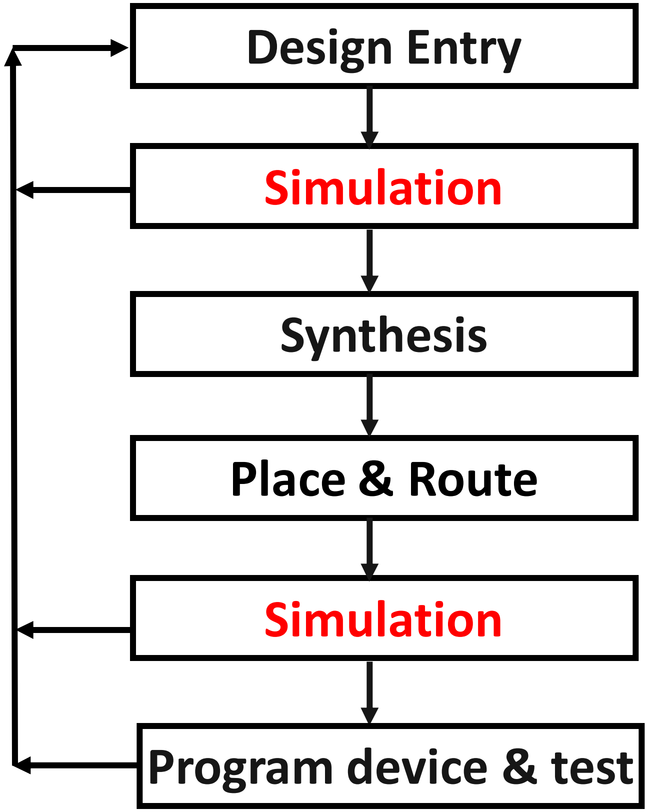
Truth Table - STUDENT:decoder24

address

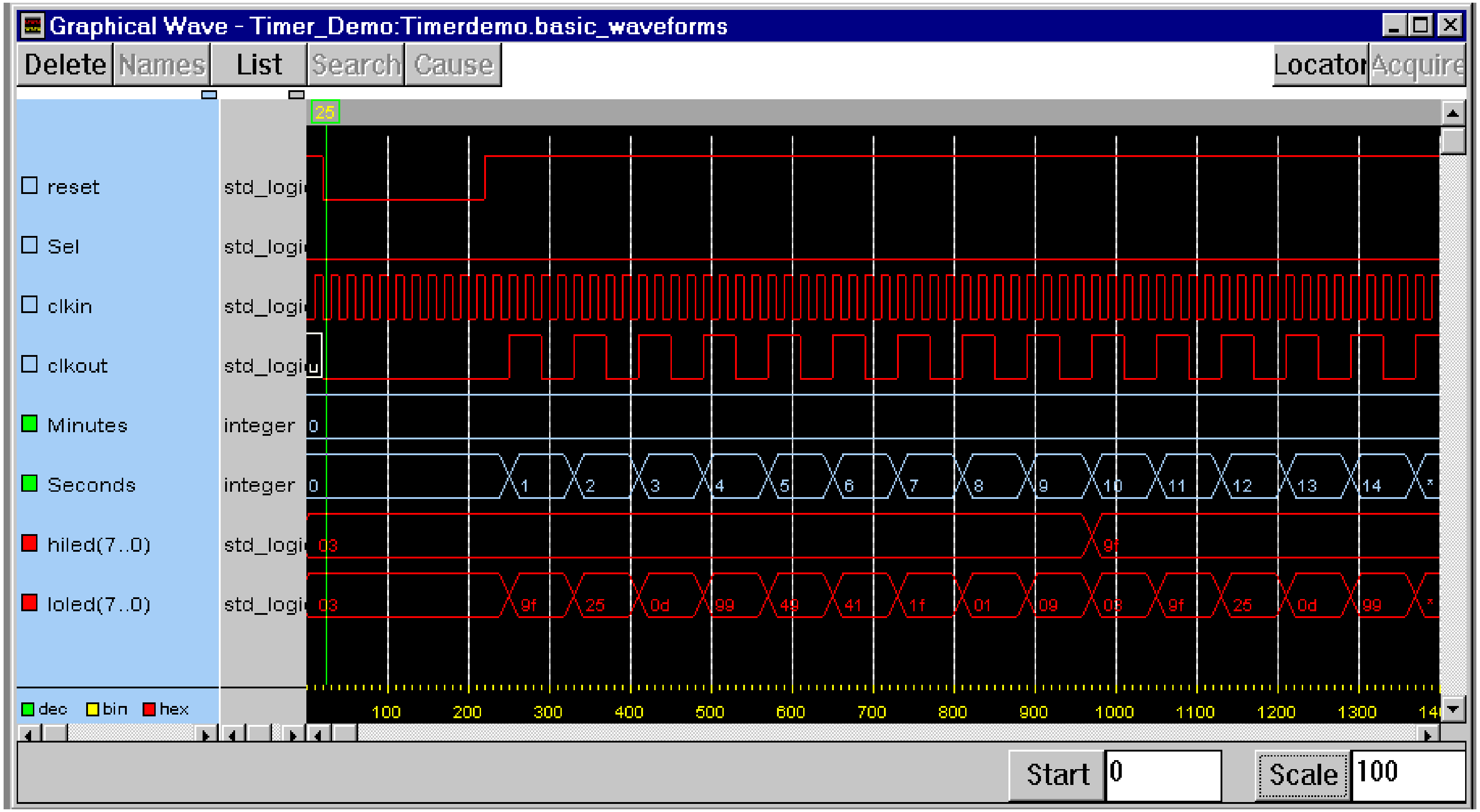
All	1	2	3	4	5	6	7
All	address	enable	data(0)	data(1)	data(2)	data(3)	
1			NOT level	NOT level	NOT level	NOT level	
2							
3	"00"	'1'	level				
4	"01"	'1'		level			
5	"10"	'1'			level		
6	"11"	'1'				level	
7		'0'	'Z'	'Z'	'Z'	'Z'	
8							

Design Cycle: Simulation

- Functional simulation:
  - simulate independent of FPGA type
  - may postpone selection
  - no timing
- Timing simulation:
  - simulate after place and routing
  - detailed timing

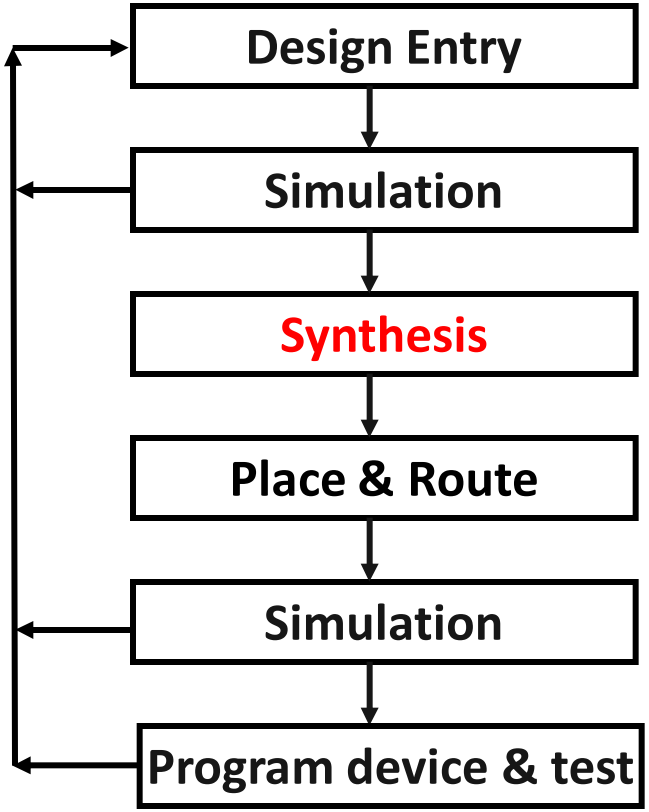


Example of simulation waveforms.
Test vectors are normally defined in a VHDL unit (testbench)



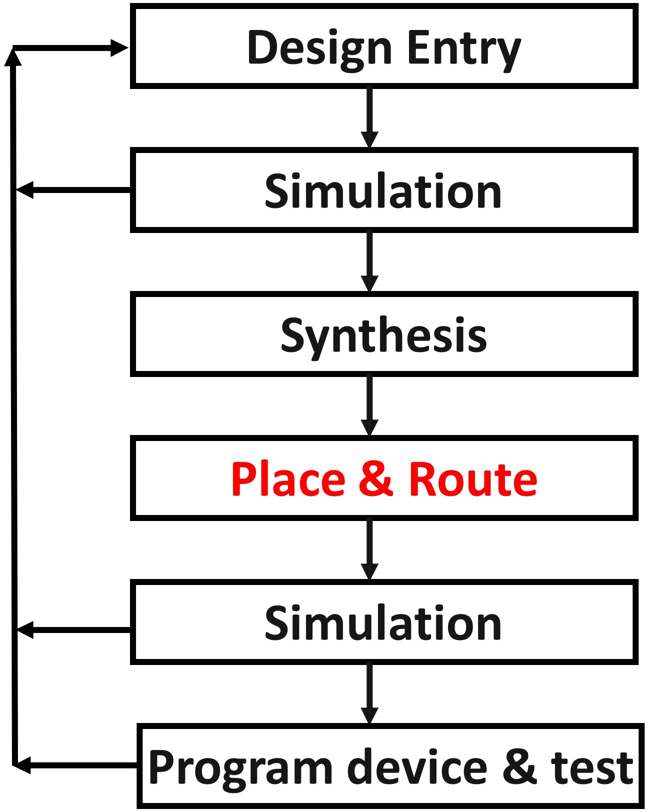
RTL Synthesis

- Input is RTL code
- Compilation & translation
  - Generates technology independent netlist
  - RTL schematic (HDL code analysis)
- Technology mapping
  - Mapping to technology specific structures:
    - **Look-up tables (LUT)**
    - **Registers**
    - **RAM/ROM**
    - **DSP blocks**
    - **Other device specific components/features**
- Logic optimization
  - Implementation analysis (technology view)



Design Cycle: Place and Route

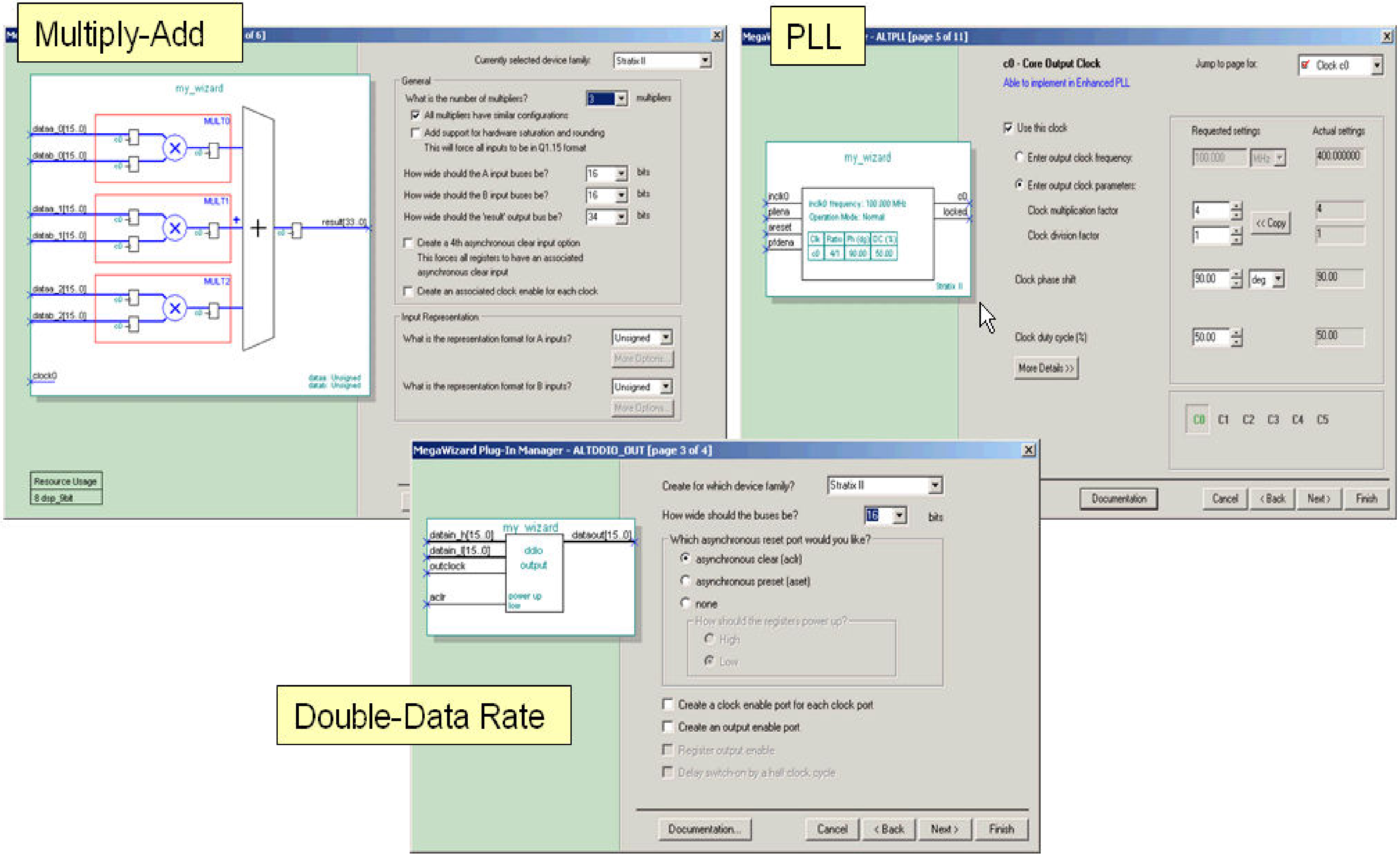
- FPGA fitter
  - Tools supplied by the FPGA vendor
  - Specific for each FPGA device architecture
- Functions
  - Place-and-route
  - Constraints editor
  - Backannotated netlist for timing simulation
  - Configuration bitstream



Macros & IP Cores

- Macros:
  - Generic pre-made design blocks:
    - e.g. PLL, FIFOs, DDR I/O, Multiply-accumulate, etc.
  - Accelerate design entry and verification
  - Pre-optimized for FPGA vendor architecture
  - Provided at no cost by the FPGA vendor to optimize performance
  - Instantiate block in the design:
    - Makes HDL code technology dependent
- IP cores:
  - More complex blocks: PCI-X interface, CPU, etc.
  - Some are provided by the FPGA vendor
  - IP cores from third party suppliers cost money
  - Evaluation before buying usually possible

Macro Example



Multiply-Add

Currently selected device family: Stratix II

General

What is the number of multipliers? 3 multipliers

☑ All multipliers have similar configurations

☐ Add support for hardware saturation and rounding
This will force all inputs to be in Q1.15 format

How wide should the A input buses be? 16 bits

How wide should the B input buses be? 16 bits

How wide should the 'result' output bus be? 34 bits

☐ Create a 4th asynchronous clear input option
This forces all registers to have an associated asynchronous clear input

☐ Create an associated clock enable for each clock

Input Representation

What is the representation format for A inputs? Unsigned

What is the representation format for B inputs? Unsigned

PLL

MegaWizard ... - ALTPLL [page 5 of 11]

c0 - Core Output Clock
Able to implement in Enhanced PLL

Jump to page for: Clock c0

☑ Use this clock

○ Enter output clock frequency:

● Enter output clock parameters:

	Requested settings	Actual settings
Enter output clock frequency	100.000 MHz	400.000000
Clock multiplication factor	4	4
Clock division factor	1	1
Clock phase shift	90.00 deg	90.00
Clock duty cycle (%)	50.00	50.00

MegaWizard Plug-In Manager - ALTDDIO_OUT [page 3 of 4]

Create for which device family? Stratix II

How wide should the buses be? 16 bits

Which asynchronous reset port would you like?

● asynchronous clear (aclr)

○ asynchronous preset (aset)

○ none

How should the registers power up?

○ High

● Low

☐ Create a clock enable port for each clock port

☐ Create an output enable port

☐ Register output enable

☐ Delay switch-on by a half clock cycle

Double-Data Rate

System On a Programmable Chip

- Many ready-made blocks for free
  - RAM/FIFO
  - UART
- Can buy ready-made parts, just like IC's: *IP Cores*
  - PCI interface
  - Processors (8051-style up to RISC/ARM processors)
- FPGA's with extra dedicated hardware built-in
  - Gigabit serialiser
  - high-end processor with RAM
- Handle different I/O standards
  - LVDS, LVPECL, LVCMOS, LVTTL, PCI, PCI-X
  - Programmable slew-rate, termination resistors

Trend in design tools

- Trying to design at a higher level of abstraction
- Starting from C/C++ or SystemC code
- Electronics for non-electronicians?

Sources of information

- CERN technical training ELEC 2005: http://indico.cern.ch/conferenceDisplay.py?confId=62928
  - Covers a lot of subjects, including optical links, EMC, description of experiments readout systems
  - Includes a lot of references to books
- LEB/LECC/TWEPP workshops from last 12 years: http://lhc-electronics-workshop.web.cern.ch/lhc%2Delectronics%2Dworkshop/
  - Detailed presentations and wider plenary talks
- PH-ESE seminars: http://indico.cern.ch/categoryDisplay.py?categId=1591
- Previous summer student lectures
- LHC (ATLAS and CMS) upgrades: ACES meeting

  http://aces.web.cern.ch/aces/