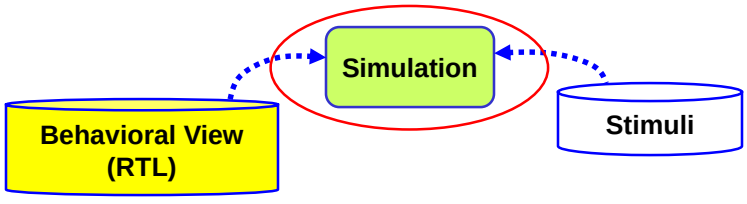
Electronic Design Automation
CSE 215

Lecture 17
High-Level Design – Verification

Mohamed Dessouky
Integrated Circuits Laboratory
Ain Shams University
Cairo, Egypt
Mohamed.Dessouky@eng.asu.edu.eg

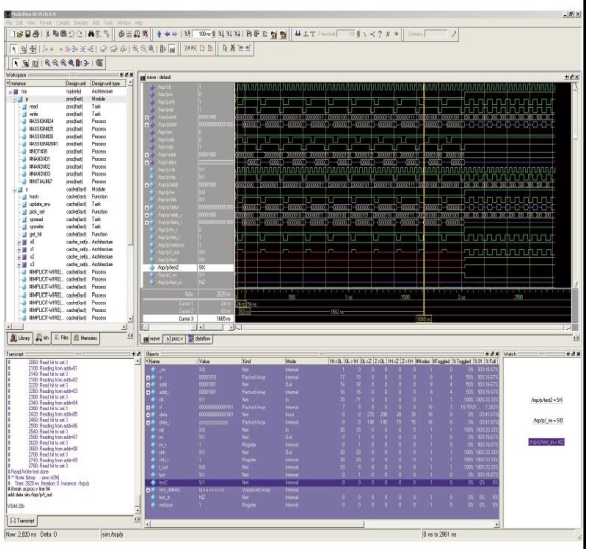
HDL Verification

- Check that the behavioral RTL do really "fulfill" user's requirements.
- Methodology depends on the design complexity:
  I. HDL simulator/Verification Environment
  II. FPGA Prototyping
  III. Emulation



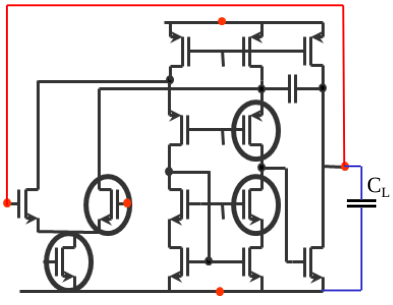
(I) Logic Simulation

- ModelSim: Industry #1 standard
- Student version download

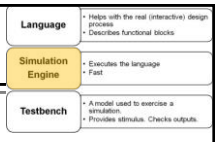


Analog Simulation

- Analog simulators must solve the entire analog system matrix (KVL & KCL) at every time step.
- Each element in the analog design can have an instantaneous influence on any other element in the matrix.
- Relies on feedback paths. There is no obvious signal flow in any direction.
- Time is continuous rather than discrete.
- Behavior heavily depends on circuit loading.



Event-driven Simulation

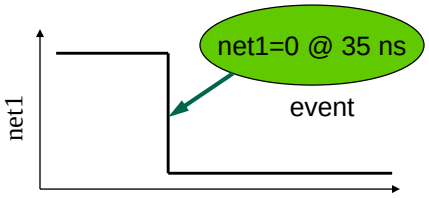


- Typically, in a digital system, around 50% of the nets would keep their value unchanged on any given clock cycle.
- Event-driven simulation is designed for digital circuits:
  - small number of signal values;
  - relatively sparse activity over time.
- Event-driven simulators try to update only those signals which change in order to reduce CPU time requirements.

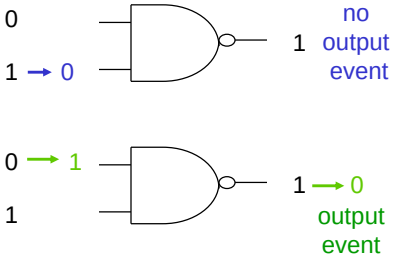
- An event is a change in a signal value.

Events

- An event is a *change* in a net's value.
- An event has two components:
  - value;
  - time.

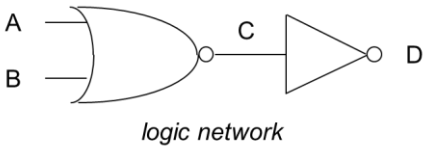


- Propagate events through simulation.
- Only when nets change value.
- Don't simulate a block until its inputs change.
- If an input change doesn't cause an output change, no event is propagated.

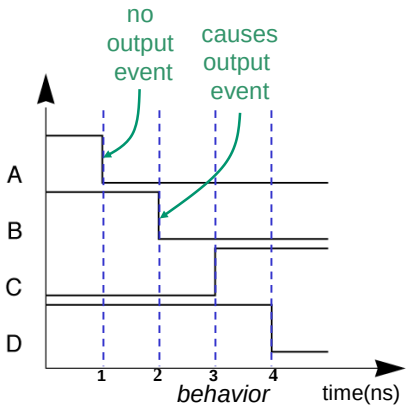


Event Propagation

- Simulator traces structure of circuits to determine causality of events—event at input of one gate may cause new event at gate's output.
- Events at primary inputs:
  - A changes at t=1;
  - B changes at t=2.
- Immediate causality:
  - C changes after 1ns delay at t=3.
- Event propagation:
  - D changes after 1ns delay at t=4.

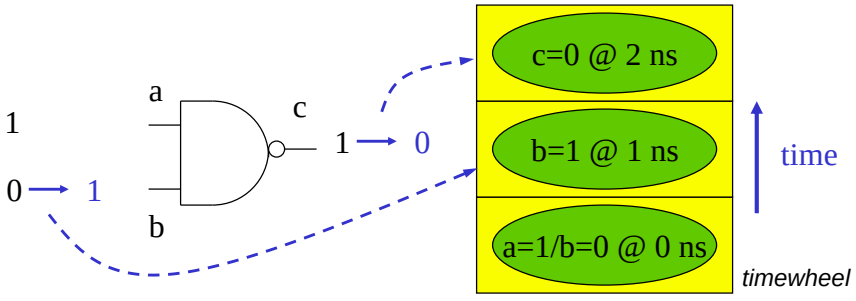


logic network

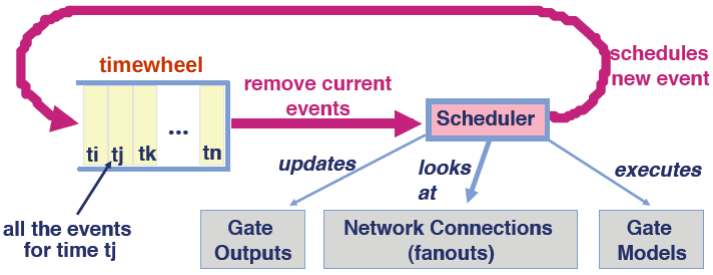


Timewheel

- A timewheel is a queue of events.
- The timewheel is a data structure in the simulator that efficiently determines the order of events processed.
- Events are placed on the timewheel in time order.
- Events are taken out of the head of the timewheel to process them in order.



Scheduler



1. Removes events from the timewheel for a given time
2. Propagates values
3. Executes models
4. Creates new events and store at appropriate order in the timewheel according to model delays and fanouts (loading)
5. Go to (1)

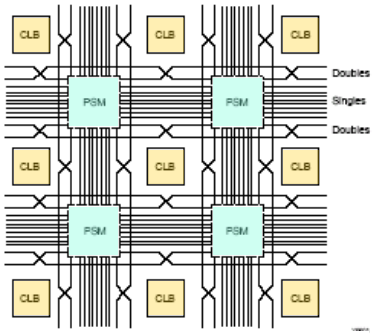
HDL Verification

- Check that the behavioral RTL do really "fulfill" user's requirements.
- Methodology depends on the design complexity:
  I. HDL simulator/Verification Environment
  II. FPGA Prototyping
  III. Emulation

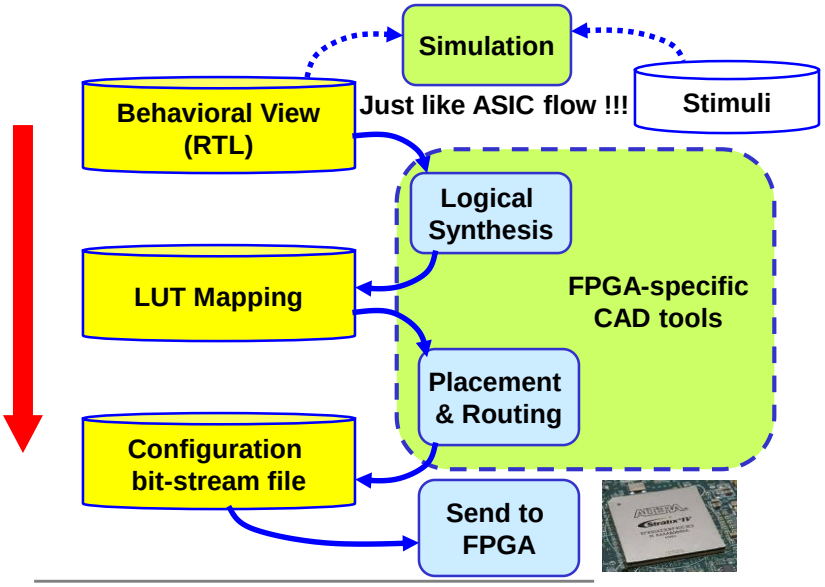
FPGA development board

Field-Programmable Gate Arrays-FPGAs

- Basic idea: two-dimensional array of logic blocks and flip-flops with a means for the user to configure (program):
  - The function of each block (**CLB: Configurable Logic Block**).
  - The interconnections between the logic blocks (**PSM: Programmable Switch Matrices**),
- Technical viewpoint:
  - For hardware/system-designers, just like ASICs, only better!
  - Fabricate a new chip every few hours.
  - Re-configurability
  - In-field Re-programmability
- At the expense of
  - Performance (speed), Area and Power.
  - Versus ASICs: delay 3-4X, area 40X, and power 12X !!



FPGA Design Flow

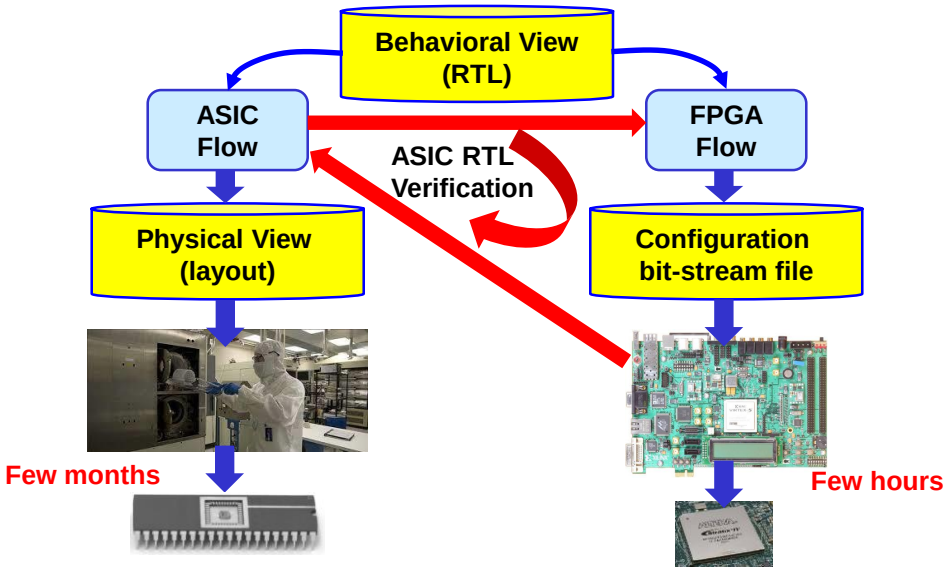


FPGA Prototyping

- Simulation speed and modeling accuracy limitations hinder hardware verification.
- FPGA prototyping is a method to prototype SoC and ASIC design on FPGA for hardware verification.
- Load design onto FPGA device (cable connects PC to development board)
- Optional “logic scope” on FPGA to access internal points
- Software development:
  - Due to time constrains, many projects cannot wait until the silicon is back from the foundry to start on software tests.
  - FPGA prototyping allows for much more time in area of software development and testing at the software-hardware integration stage.

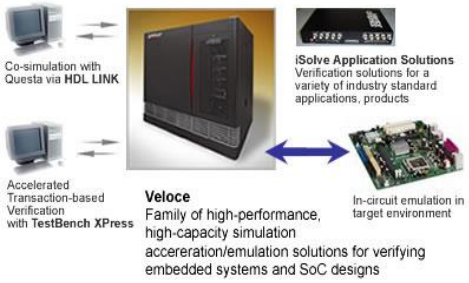
FPGA development board

ASIC *vs.* FPGA Flow



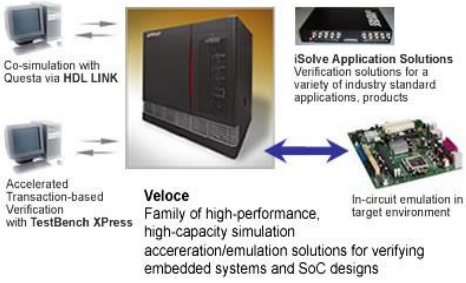
HDL Verification

- Check that the behavioral RTL do really “fulfill” user’s requirements.
- Methodology depends on the design complexity:
  I. HDL simulator/Verification Environment
  II. FPGA Prototyping
  III. Emulation



(III) Hardware Emulation

- A special hardware of many FPGAs and dedicated software for real-time verification.
- Testbench runs on the computer, while the DUT runs on the FPGA.
- Imitating the behavior of a system under design with another piece of hardware.
- The emulator is fast enough to be plugged into a working target system in place of a yet-to-be-built chip, so the whole system can be debugged with live data.
- Example: http://www.mentor.com/products/fv/emulation-systems/



Summary

- **Logic simulation:**
  - Easy, accurate, flexible, and low cost.
  - However, not fast enough for large designs.
  - Too slow to run application software against the hardware design.
- **FPGA-based prototypes:**
  - Fast and inexpensive.
  - Long time required to implement a large design.
  - Little debugging capability.
- **Emulation:**
  - Improves greatly on FPGA prototyping's long time.
  - Provides a comprehensive, efficient debugging capability.
  - Test application software + hardware debug environment.
  - At the expense of running speed and cost compared to FPGA.

References

- Franck Wajsbürt and Jean-Paul Chaput, "Outils de CAO pour VLSI - Flot de conception VLSI Alliance", University of Paris VI.
- Wayne Wolf, "Modern VLSI Design", Third Edition, PEARSON, Prentice Hall