FYS4220/9220

UNIVERSITY OF OSLO

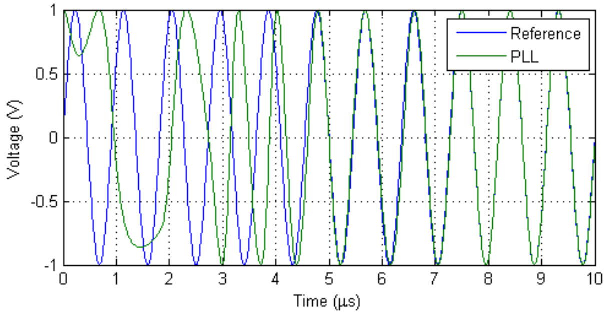
Digital design & Embedded systems

Lecture #5

J. K. Bekkeng, 2.7.2011

Phase-locked loop (PLL)

- Implemented using a VCO (Voltage controlled oscillator), a phase detector and a closed feedback loop
- A PLL generates a signal where
  phase and frequency follows the input signal (”referansen”)



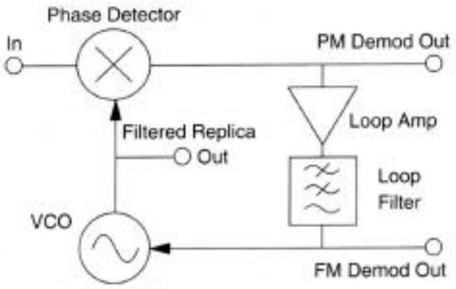


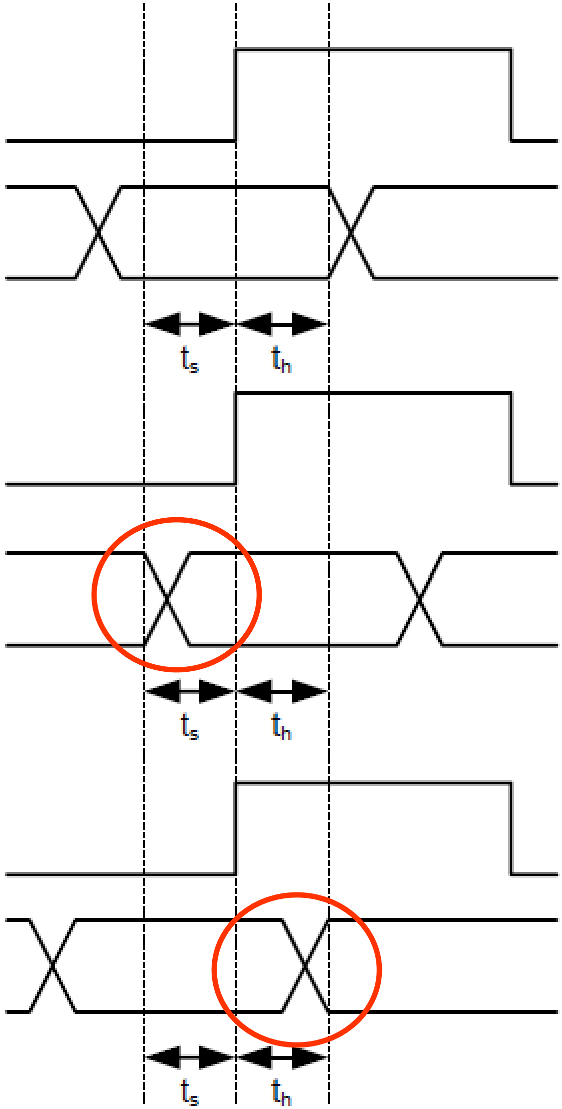
Figure 15.17. Phase-locked loop.

- Use in digital design:
  - Create a clock which is synchronized/phase locked to another clock or data signal
  - Clock multiplication and division (clock divider)
- FPGA’s typically have several PLLs

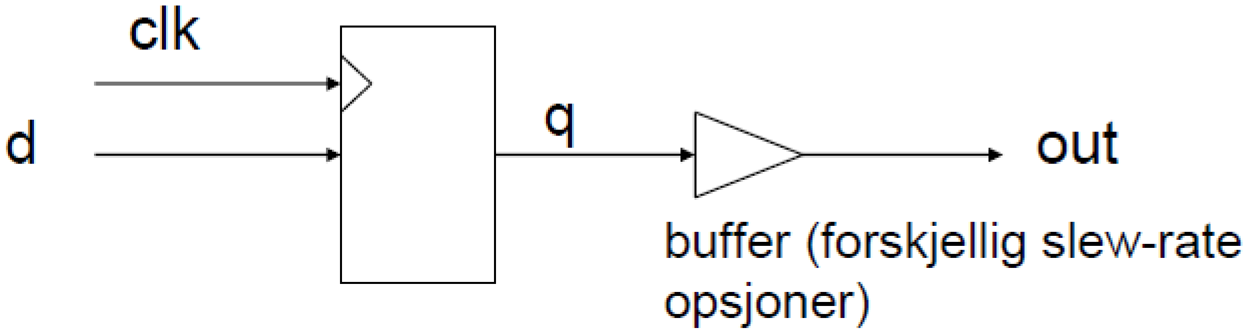
Set-up/hold times

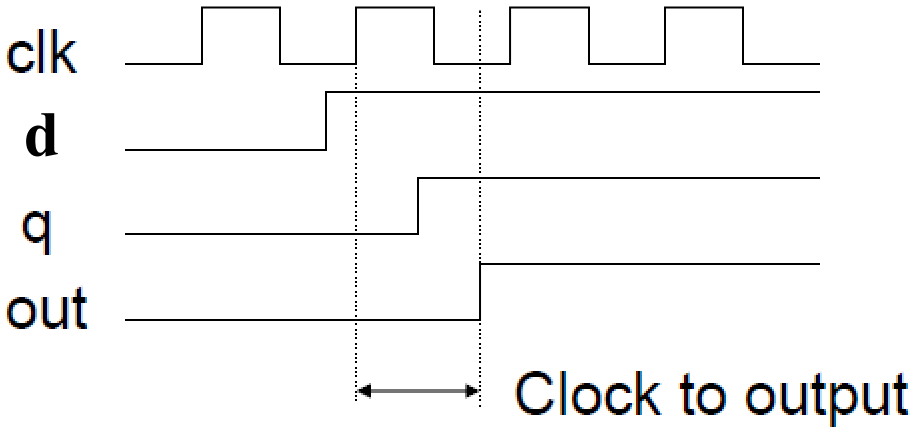
- The input must be stable for a certain time before (set-up time) and after (hold time) a clock edge, to avoid metastability (that the output goes to a metastable state – which is a state different from '0' or '1').

- The output will return to '0' or '1' after having been in the metastable state, but it is uncertain at which state it ends up. Therefore the system is no longer deterministic.

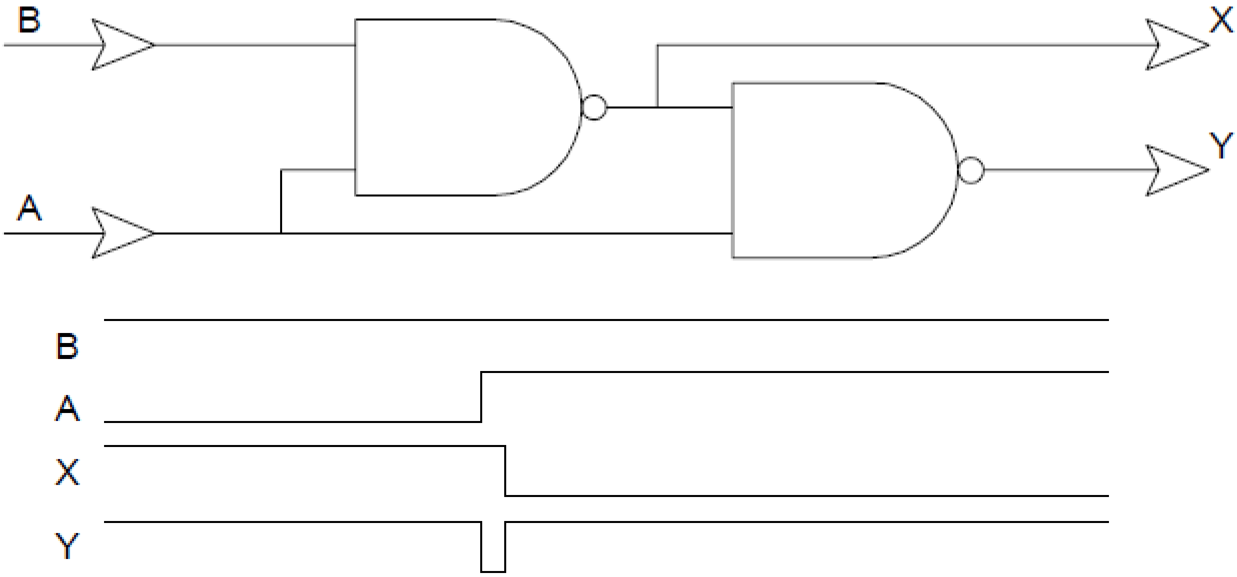


Clock to output delay



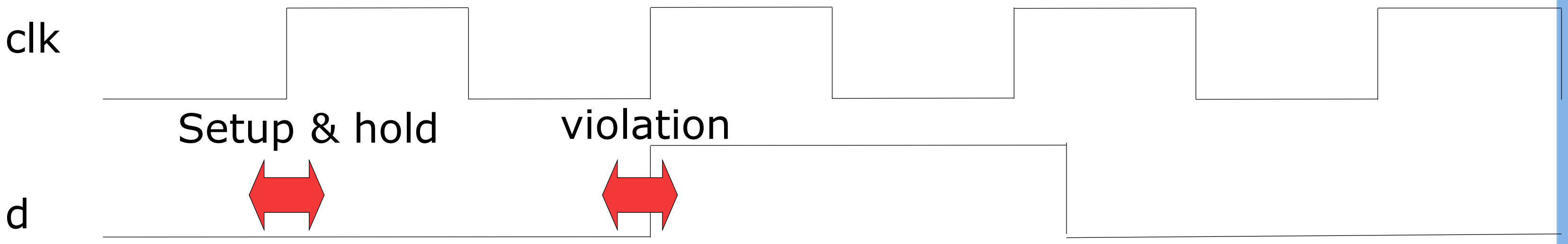


Time delay through logic gates
- use synchronous logic!



Multiple Clock Domains

- In many designs two (or more) clocks with different frequencies exist
- Data moved across clock domains appears asynchronous to the receiving (destination) domain
- Asynchronous data will cause meta-stability
- The only safe way: use a *synchronizer*

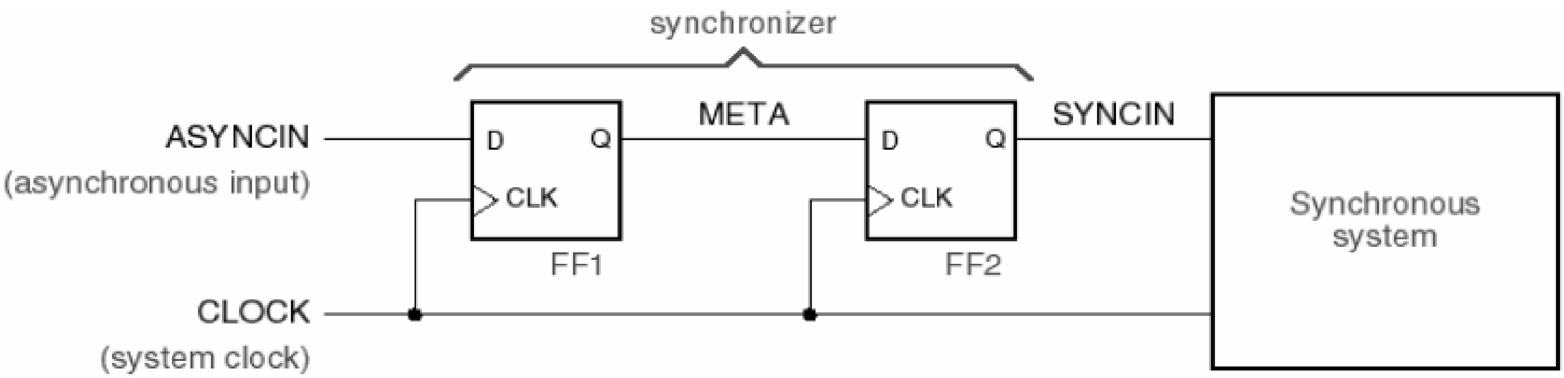


2 - Flip-Flop Synchronizer

- Most common type of (bit) synchronizer
- Eliminate output metastability
- FF1 will go metastable, but FF2 does not look at data until a clock period later, giving FF1 time to stabilize (FF = Flip Flop)

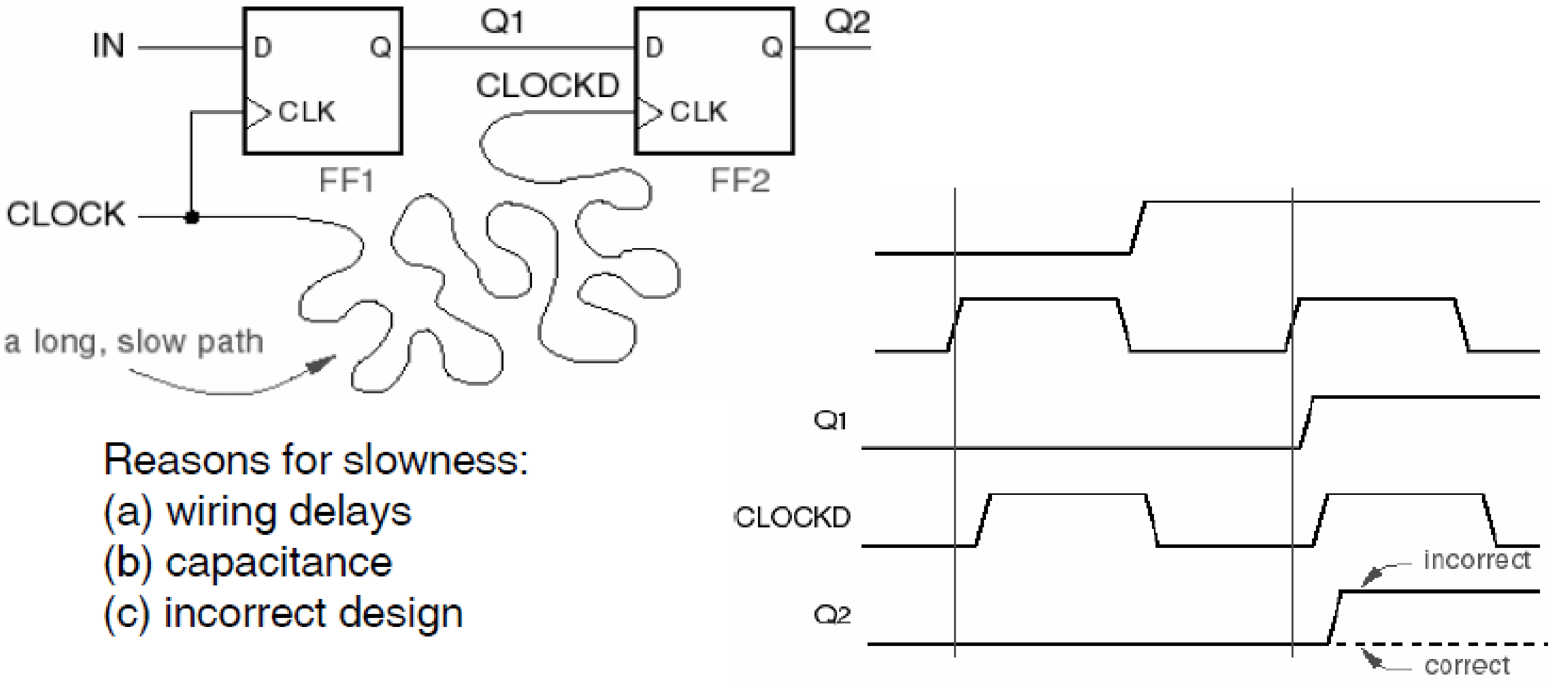
Metastable: a voltage between the high and low voltage thresholds

Recommended synchronizer design



Clock Skew

- Clock signal may not reach all flip-flops simultaneously.
- Output changes of flip-flops receiving “early” clock may reach D inputs of flip-flops with “late” clock too soon.



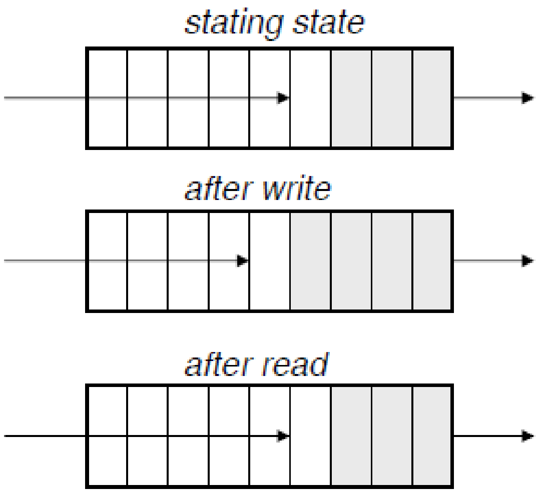
Reasons for slowness:
(a) wiring delays
(b) capacitance
(c) incorrect design

Multiple Clock Domains (MCD)

- Arise naturally in interfacing with the outside world
- Reduce power and energy consumption
- Typically different parts of the circuit run at different frequencies
- MCD synchronizer: FIFO

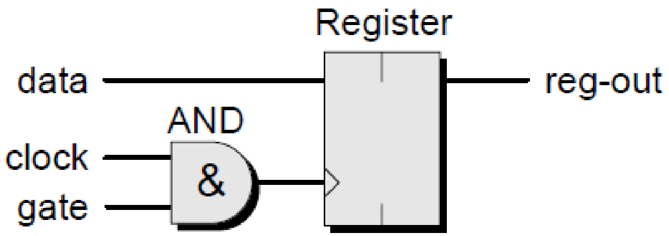
First-in-first-out (FIFO) Memory

- Used to implement *queues*.
- These find common use in computers and communication circuits.
- Generally, used for rate matching data producer and consumer:

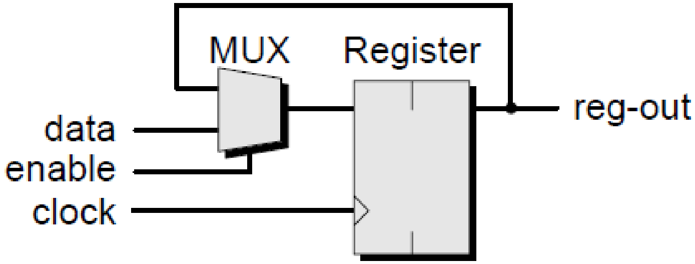


- Producer can perform many writes without consumer performing any reads (or vice versa). However, because of finite buffer size, on average, need equal number of reads and writes.
- Typical uses:
  - interfacing I/O devices. Example network interface. Data bursts from network, then processor bursts to memory buffer (or reads one word at a time from interface). Operations not synchronized.
  - Example: Audio output. Processor produces output samples in bursts (during process swap-in time). Audio DAC clocks it out at constant sample rate.

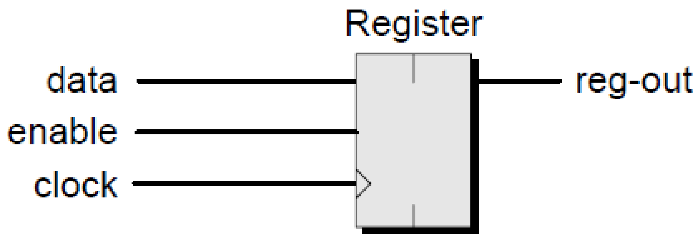
Klokkeenabling vs klokkegating



(a) Clock gating

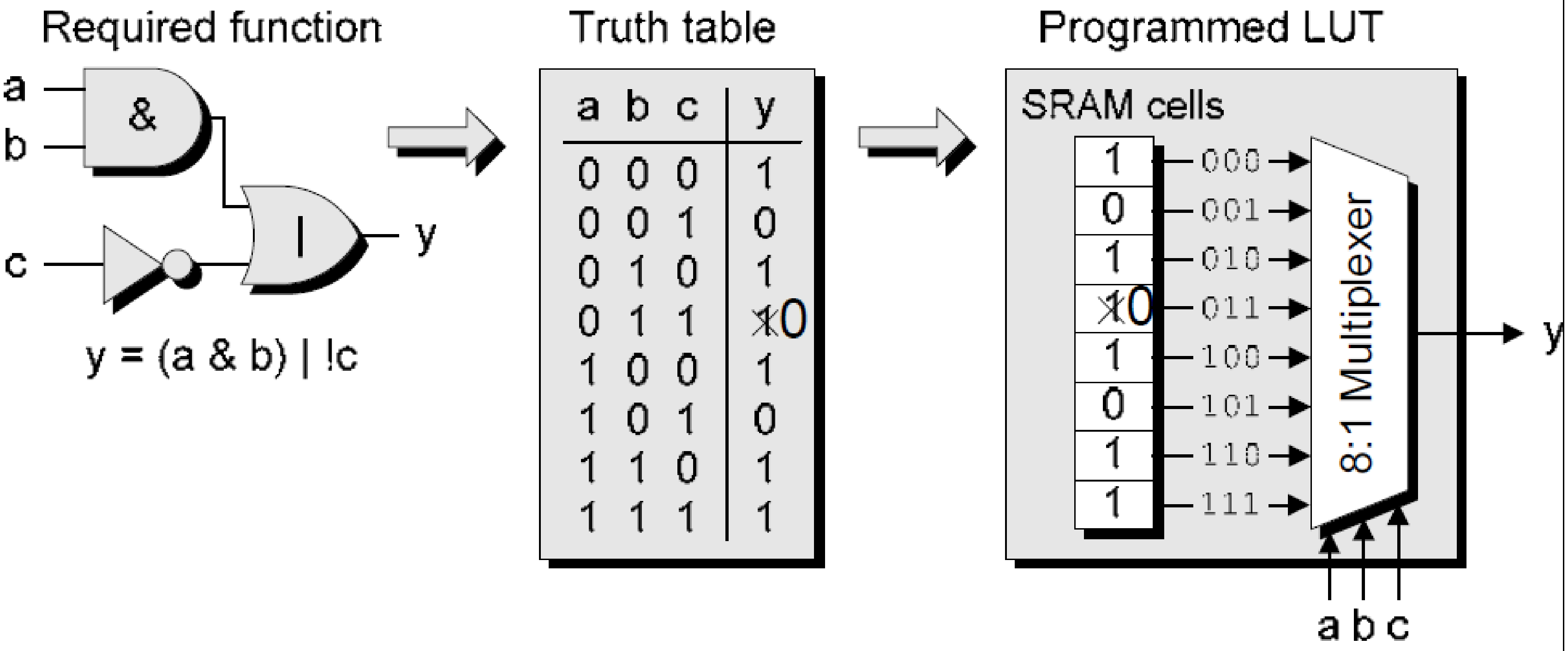


(b) Clock enabling ("then")



(b) Clock enabling ("now")

Look-Up Table (LUT)



Required function

y = (a & b) | !c

Truth table

a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	~~1~~0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

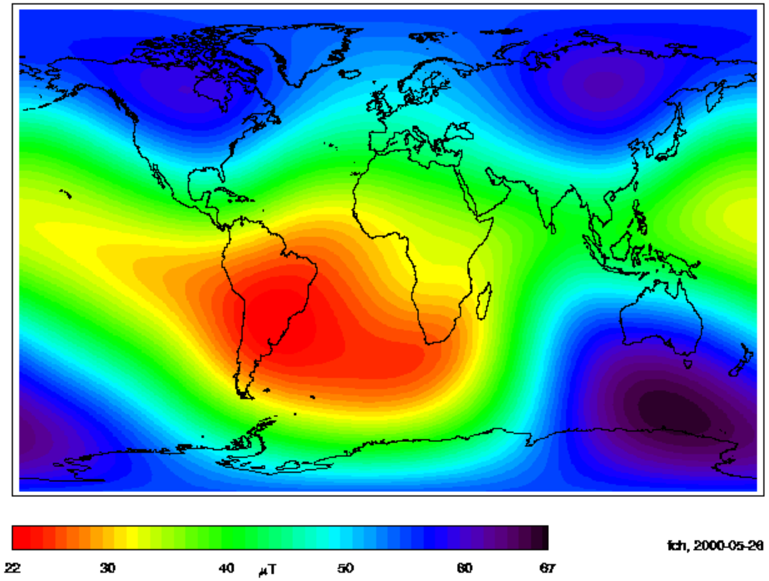
Programmed LUT

SRAM cells

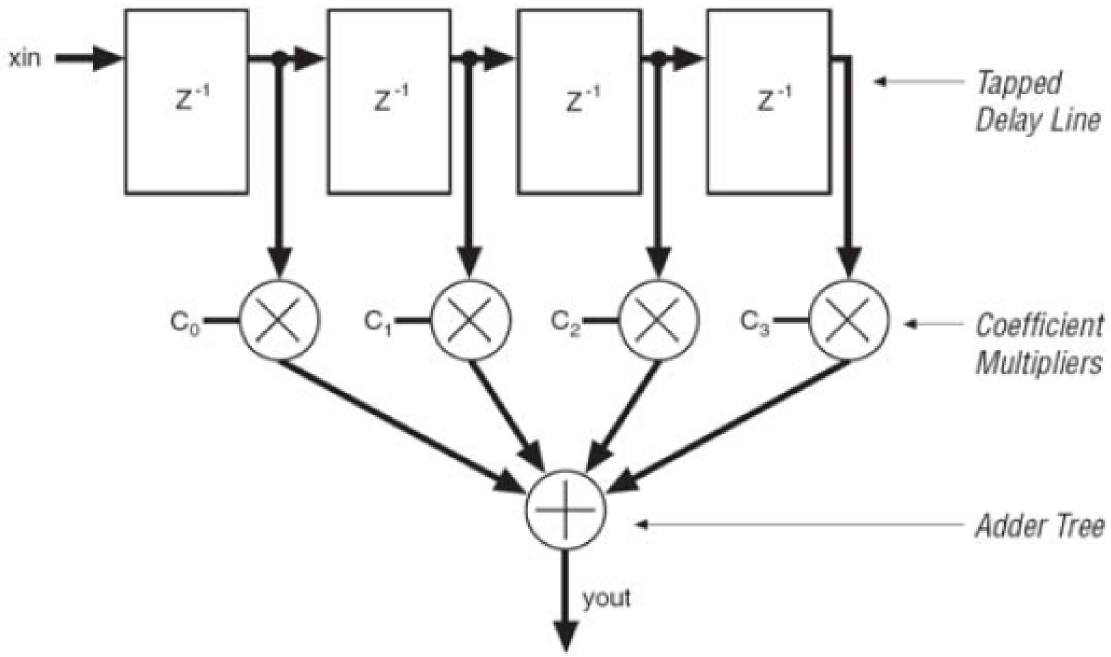
1	000
0	001
1	010
~~1~~0	011
1	100
0	101
1	110
1	111

Examples of the use of LUTs

- Store values for **Sine** and **cosine** functions
- In aerospace a LUT is usually used to store magnetic field model data (used in the navigation algorithm) because the magnetic field models (IGRF /WMM) are to computationally demanding to run in real-time on the onboard computer



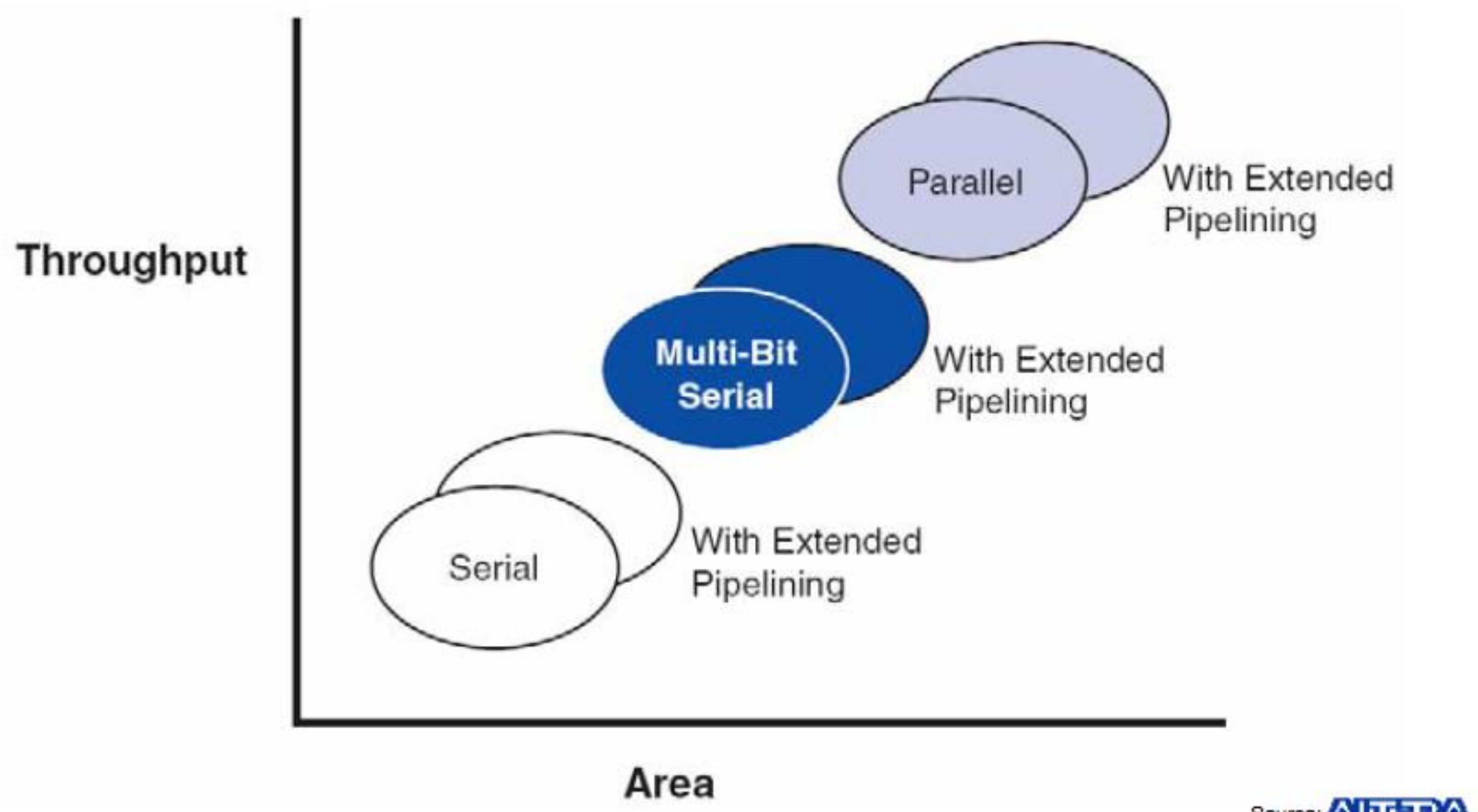
Basic FIR Filter



- Altera and Xilinx provide FIR filter compiler support.
- These examples are taken from Altera's "FIR Compiler User's Guide".

Source: ALTERA

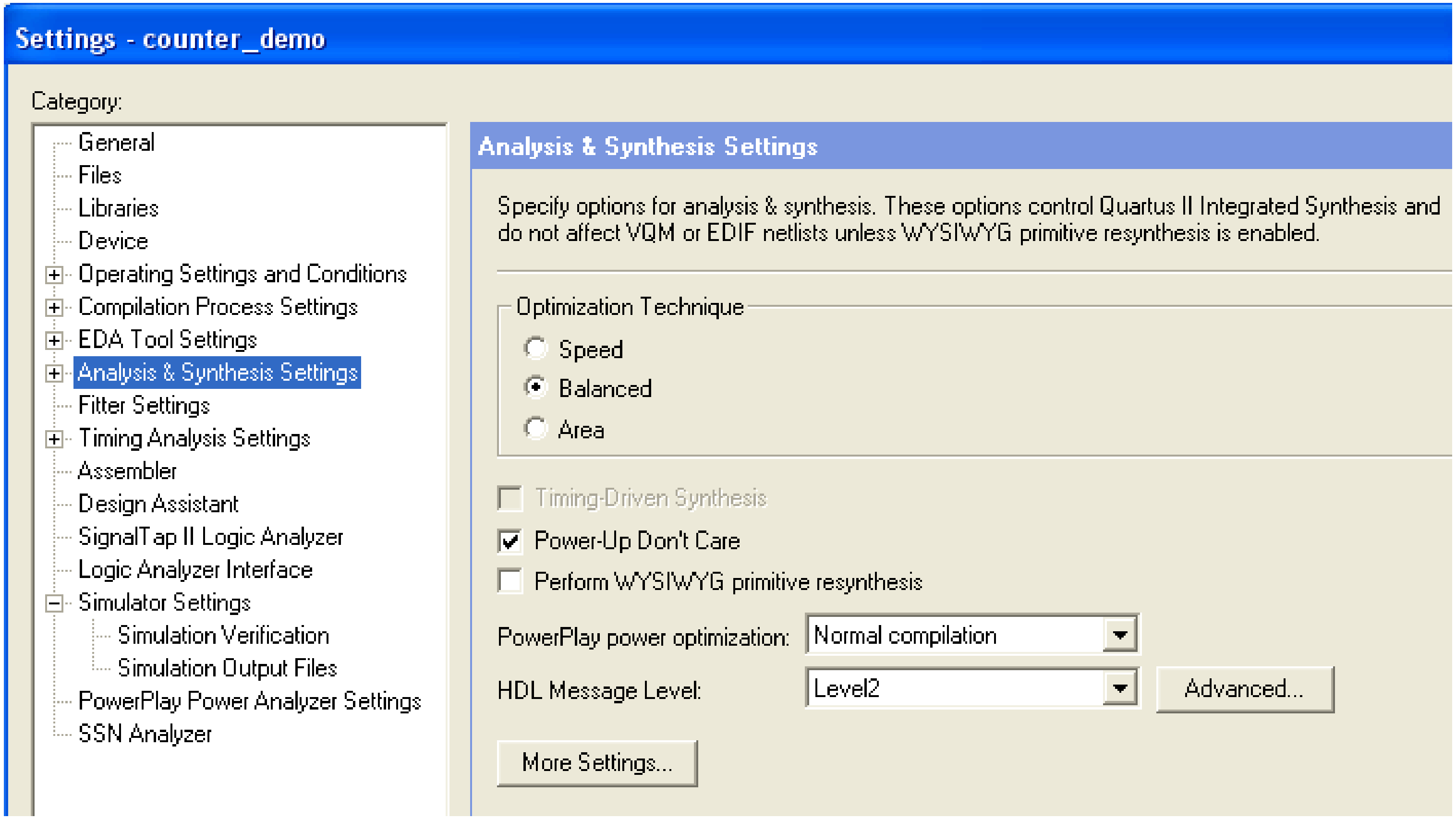
FIR Filter Compiler Design Space



Source: ALTERA

Speed vs Area optimization

Speed/Area optimization can be set in Quartus II before synthesis



Settings - counter_demo

Category:

- General
- Files
- Libraries
- Device
- Operating Settings and Conditions
- Compilation Process Settings
- EDA Tool Settings
- Analysis & Synthesis Settings
- Fitter Settings
- Timing Analysis Settings
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- Simulator Settings
  - Simulation Verification
  - Simulation Output Files
- PowerPlay Power Analyzer Settings
- SSN Analyzer

Analysis & Synthesis Settings

Specify options for analysis & synthesis. These options control Quartus II Integrated Synthesis and do not affect VQM or EDIF netlists unless WYSIWYG primitive resynthesis is enabled.

Optimization Technique

○ Speed
● Balanced
○ Area

□ Timing-Driven Synthesis
☑ Power-Up Don't Care
□ Perform WYSIWYG primitive resynthesis

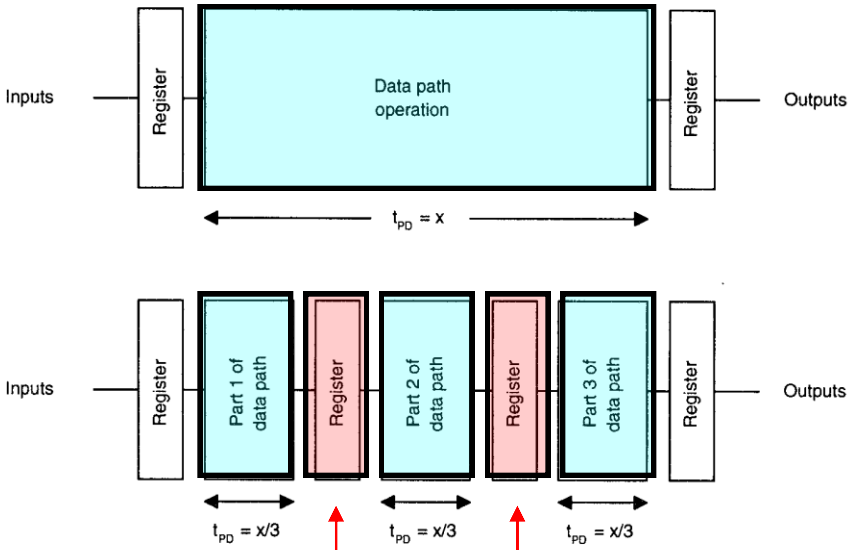
PowerPlay power optimization: Normal compilation

HDL Message Level: Level2

Pipelining

- Used to increase the **throughput** (amount of data processed in a given time period)
- Long data paths are broken into shorter data paths, such that the system frequency can be increased
- The data path is broken by adding registers in the data path

Pipelining



Breaks up the data path in three shorter data paths by using two pipelining registers

Without pipelining:

- t_{pd} = T => f_{max} = 1/T

- With 2 pipelining registers (ideal)
  - t_{pd} = T/3 => f_{max} = 3*(1/T)
  - Allows a system frequency which is three times higher

- With 2 pipelining registers (real)
  - Delay in the pipelining registers gives that the system frequency can increase less than a factor of three

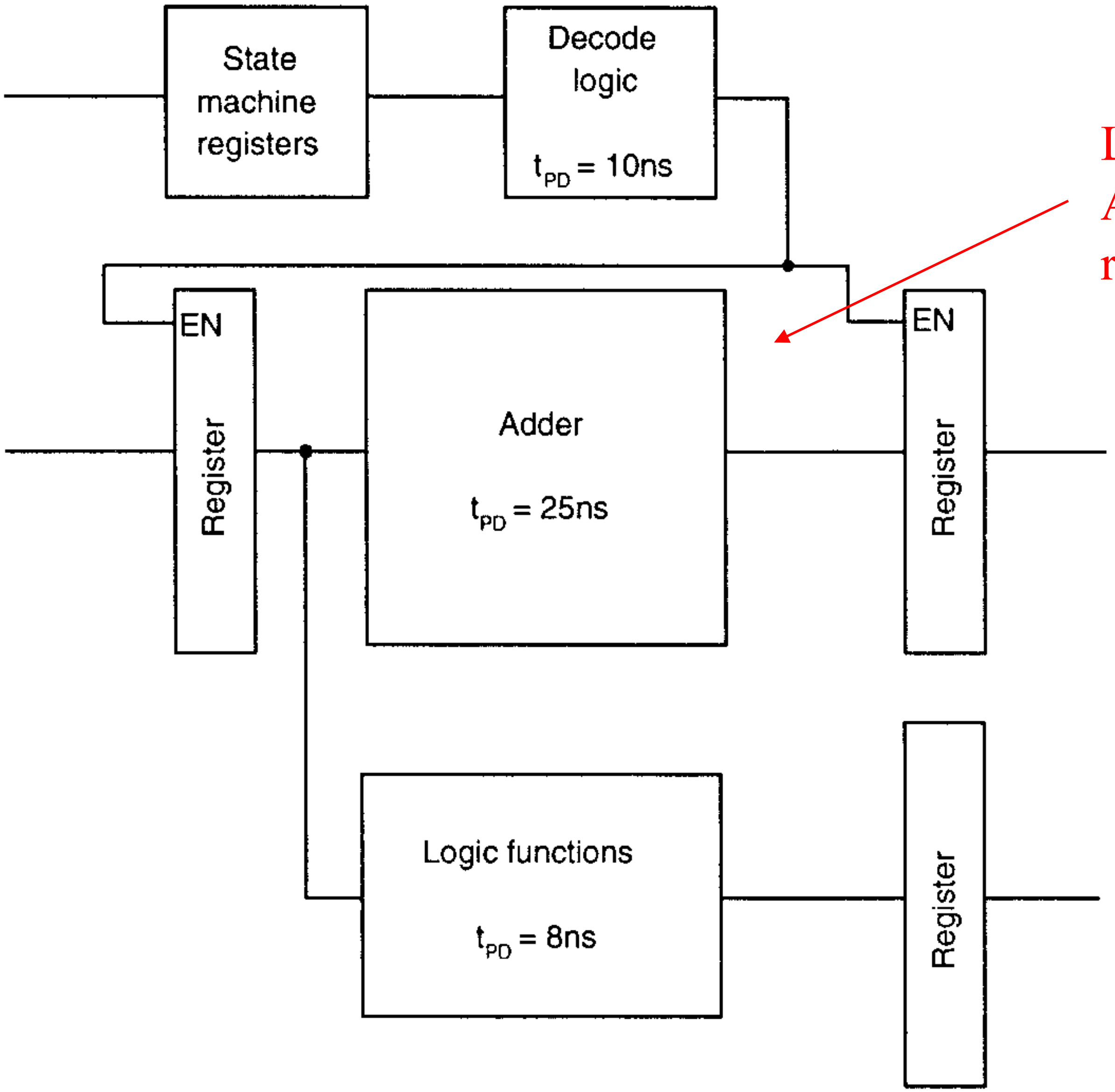
pd = propagation delay

Pipelining summary

- Pros
  - **Increased performance !**
- Cons
  - **Increased latency**
  - **Demands more logic**

Use of pipelining in programmable logic

- Used to increase the performance in FPGA-designs
- FPGA well suited for pipelining due to many flip-flops
- Pipelining is used less frequently in CPLDs
  - CPLDs can perform lager operations in a single clock cycle compared to FPGAs

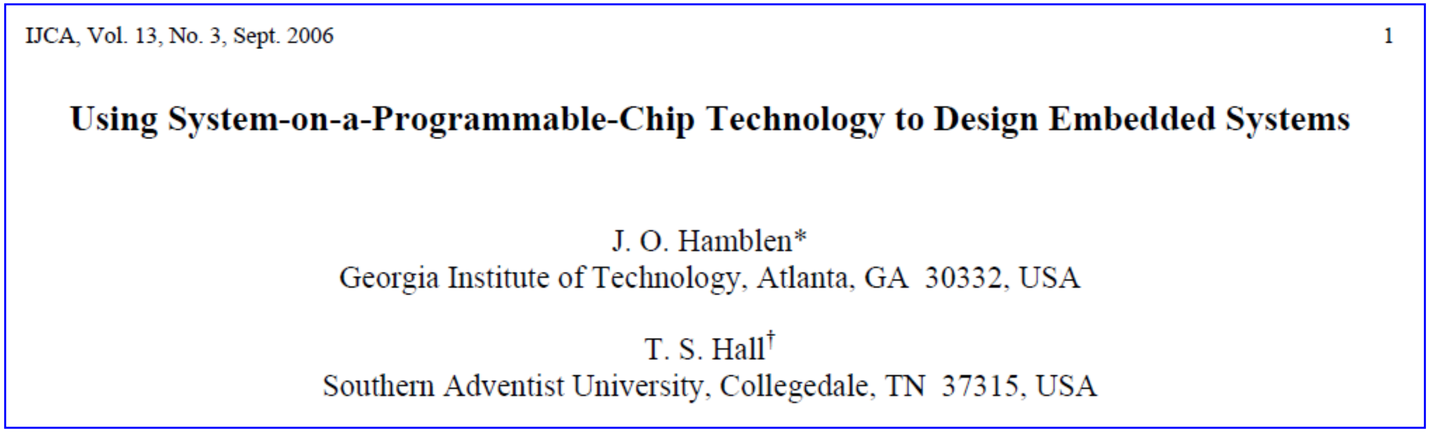


Largest delay is here! Add pipelining registers if needed!

Figure 9-11 Point-to-point delays

System-On-a-Programmable-Chip

The following paper is part of the coarse syllabus:



IJCA, Vol. 13, No. 3, Sept. 2006 1

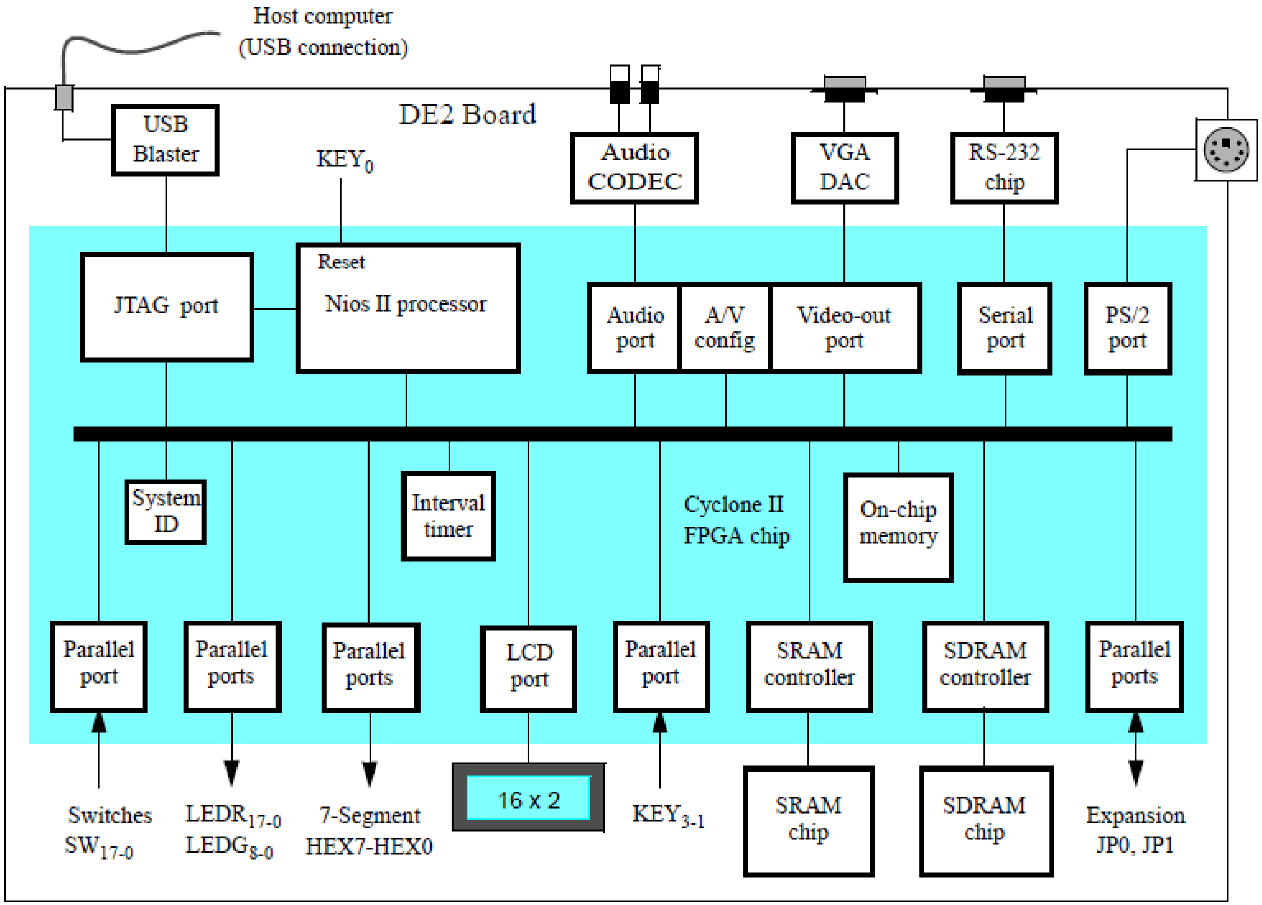
Using System-on-a-Programmable-Chip Technology to Design Embedded Systems

J. O. Hamblen*
Georgia Institute of Technology, Atlanta, GA 30332, USA

T. S. Hall†
Southern Adventist University, Collegedale, TN 37315, USA

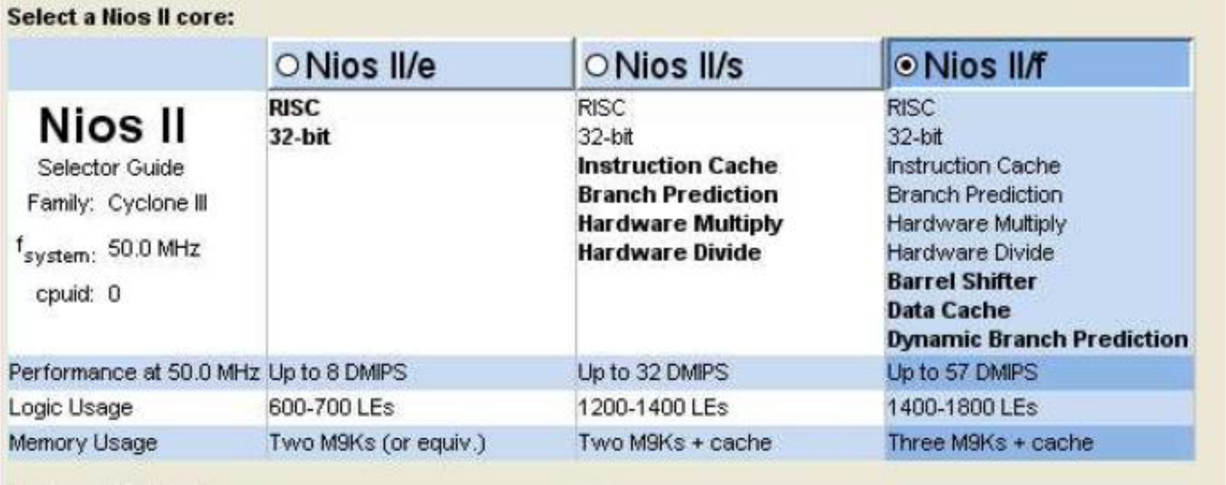
Embedded system Example

- DE2 Media computer; see the document *”Media Computer System for the Altera DE2 Boards”*



Nios II Processor

- 32-bit CPU (RISC processor)
- Three versions of the NIOS II processor are available, designated economy (/e), standard (/s), and fast (/f)
- Use either assembly language or the C programming language



Select a Nios II core:

	○ Nios II/e	○ Nios II/s	● Nios II/f
Nios II Selector Guide Family: Cyclone III f_{system}: 50.0 MHz cpuid: 0	**RISC** **32-bit**	RISC 32-bit **Instruction Cache** **Branch Prediction** **Hardware Multiply** **Hardware Divide**	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide **Barrel Shifter** **Data Cache** **Dynamic Branch Prediction**
Performance at 50.0 MHz	Up to 8 DMIPS	Up to 32 DMIPS	Up to 57 DMIPS
Logic Usage	600-700 LEs	1200-1400 LEs	1400-1800 LEs
Memory Usage	Two M9Ks (or equiv.)	Two M9Ks + cache	Three M9Ks + cache

Nios II Hardware Development Tutorial

Nios II Software Developer's Handbook

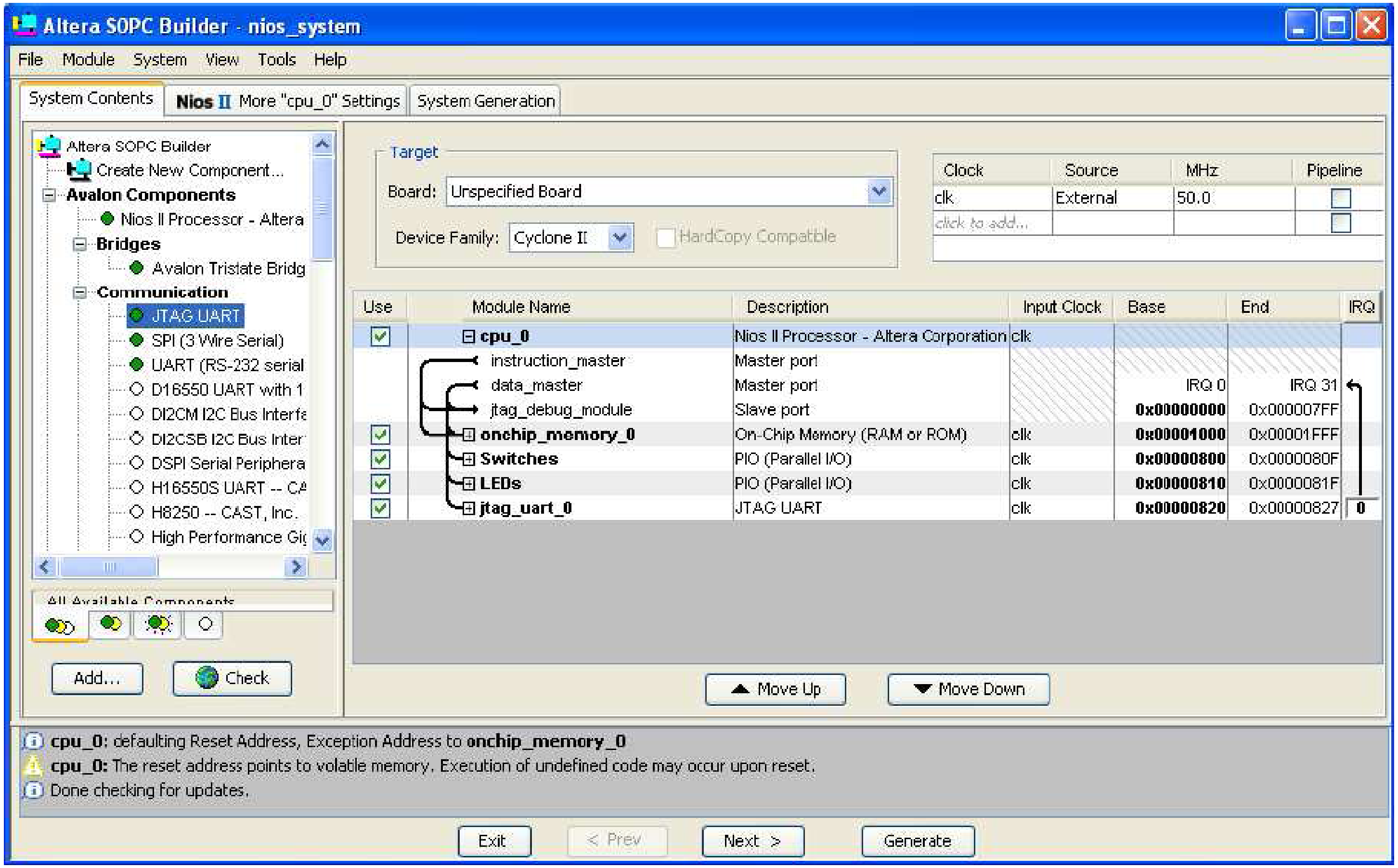
Creating Multiprocessor Nios II Systems Tutorial

Nios II Processor Reference Handbook

And many more references …..

Altera SOPC Builder

- SOPC = System-On-a-Programmable-Chip
- SOPC Builder is used to implement a system that uses the Nios II processor on an Altera FPGA device



Board: Unspecified Board

Device Family: Cyclone II

□ HardCopy Compatible

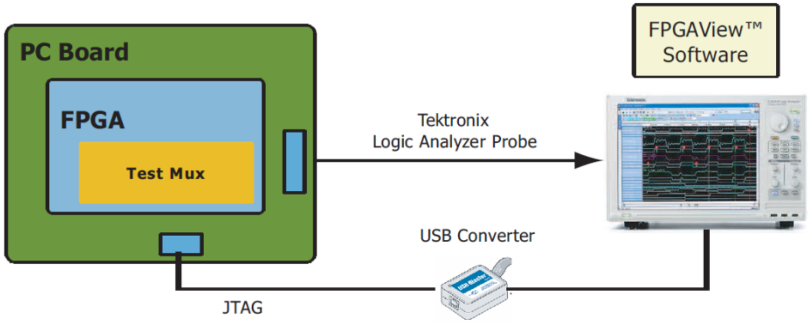
Clock	Source	MHz	Pipeline
clk	External	50.0	□
click to add...			□

Use	Module Name	Description	Input Clock	Base	End	IRQ
☑	⊟ cpu_0	Nios II Processor - Altera Corporation	clk			
	instruction_master	Master port				
	data_master	Master port		IRQ 0	IRQ 31	
	jtag_debug_module	Slave port		0x00000000	0x000007FF	
☑	⊞ onchip_memory_0	On-Chip Memory (RAM or ROM)	clk	0x00001000	0x00001FFF	
☑	⊞ Switches	PIO (Parallel I/O)	clk	0x00000800	0x0000080F	
☑	⊞ LEDs	PIO (Parallel I/O)	clk	0x00000810	0x0000081F	
☑	⊞ jtag_uart_0	JTAG UART	clk	0x00000820	0x00000827	0

```
cpu_0: defaulting Reset Address, Exception Address to onchip_memory_0
cpu_0: The reset address points to volatile memory. Execution of undefined code may occur upon reset.
Done checking for updates.
```

On-chip debugging

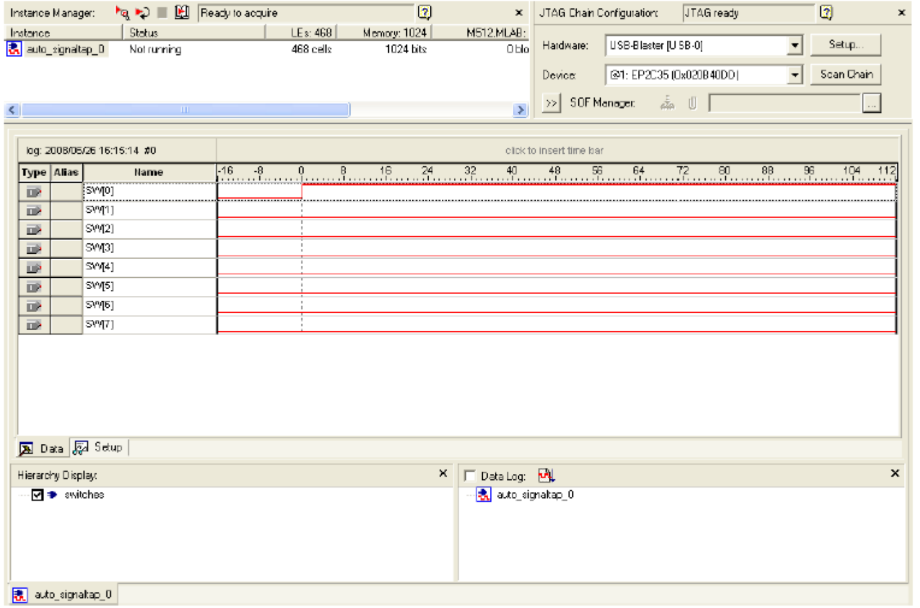
- Three possible solutions:
  1. Route internal signals to test connectors and use an external logic analyzer to look at the signals
  2. Use an on-chip debugger
  3. Use an advanced test unit, e.g. an advanced oscilloscope with a logic analyzer which supports debugging of FPGAs by JTAG (Tektronix and Agilent have such oscilloscopes)



- On-chip debuggers:
  - ChipScope (Xilinx)
  - SignalTap II (Altera)
    - Note: not all circuit families are supported!

On-chip debugging

- “**The SignalTap II Embedded Logic Analyzer** is scalable, easy to use, and is **included with the Quartus II software subscription**. This logic analyzer helps debug an FPGA design by **probing the state of the internal signals in the design without the use of external equipment”.**
- “The SignalTap II Embedded Logic Analyzer **does not require external probes** or changes to the design files to capture the state of the internal nodes or I/O pins in the design.”



Instance Manager: Ready to acquire

Instance	Status	LEs: 468	Memory: 1024	M512,MLAB:
auto_signaltap_0	Not running	468 cells	1024 bits	0 blo

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0]

Device: @1: EP2C35 (0x020B40DD)

SOF Manager:

log: 2008/06/26 16:15:14 #0

Type	Alias	Name
		SW[0]
		SW[1]
		SW[2]
		SW[3]
		SW[4]
		SW[5]
		SW[6]
		SW[7]

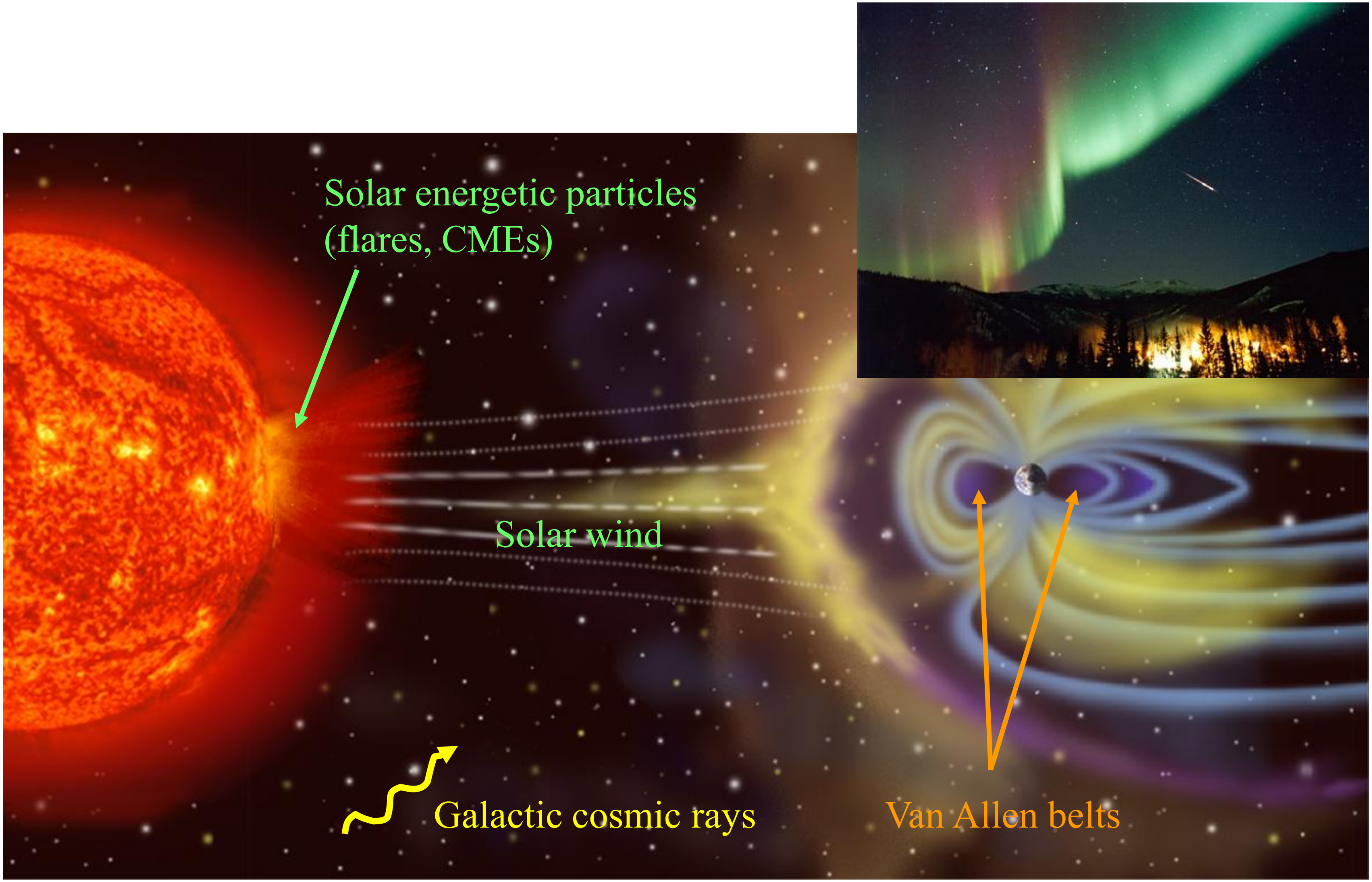
Hierarchy Display:

☑ switches

□ Data Log:

auto_signaltap_0

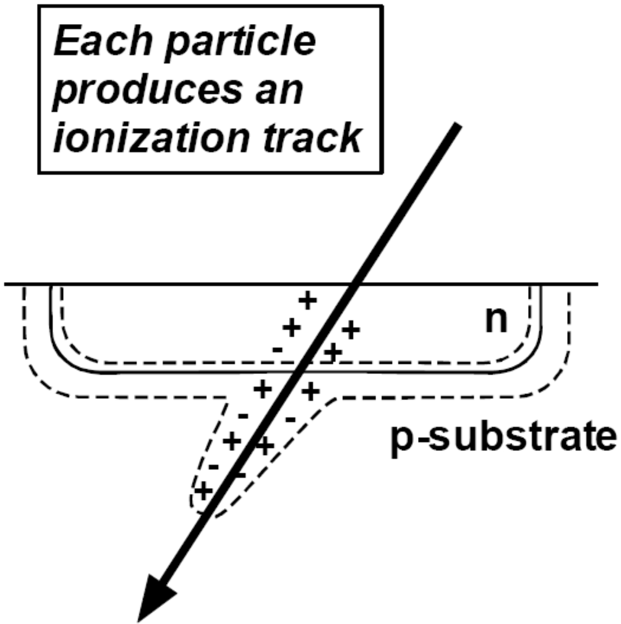
Space Radiation Environment



Radiation Effects on Electronics

- Total ionizing dose (TID) effects
  - Accumulation of ionizing dose deposition over a long time.
- Displacement damage (DD)
  - Accumulation of crystal lattice defects caused by high energy radiation.
- Single event effects (SEE)
  - A high ionizing dose deposition, from a single high energy particle, occurring in a sensitive region of the device.

Single Event Effects



- Single event upset (SEU)
- Single event transient (SET)
- Single event latchup (SEL)
- Single event burnout (SEB)

NMOS transistor

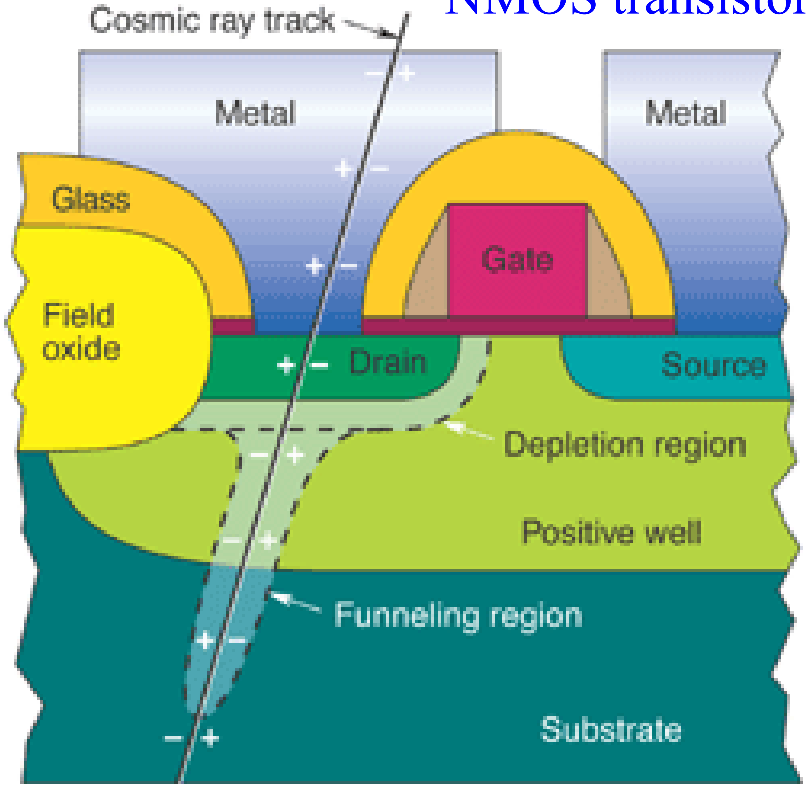


Illustration by J. Scarpulla et al.

Single event upset (SEU)

- Internal charge deposition causes a ”bit flip” in a memory element or change of state in a logic circuit.
- SEU occurs in e.g. computer memories and microprocessors.

↓

Possibly non-Destructive effects:

• Corruption of the information stored in a memory element.

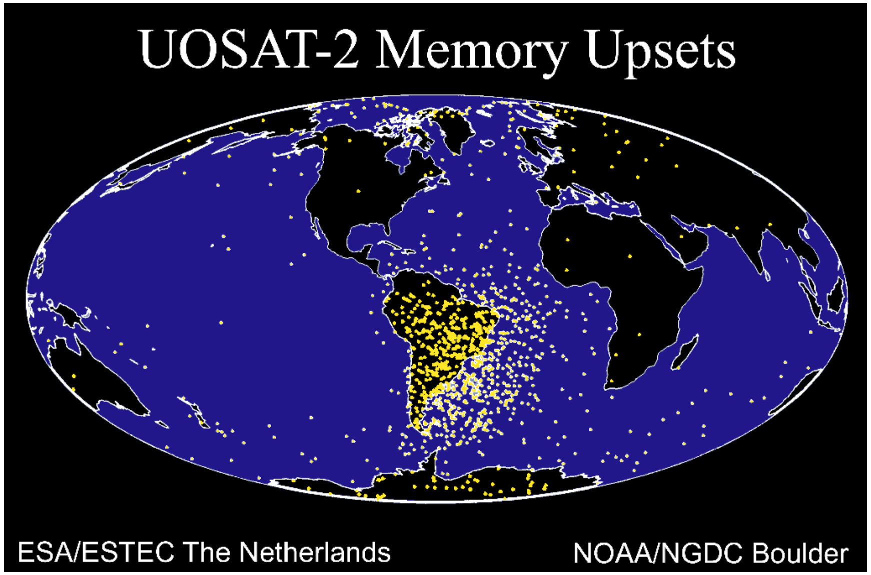
• Usually not permanent damage; a memory element/logic state can be refreshed with a new/correct value if the SEU is detected.

Possibly destructive effects:

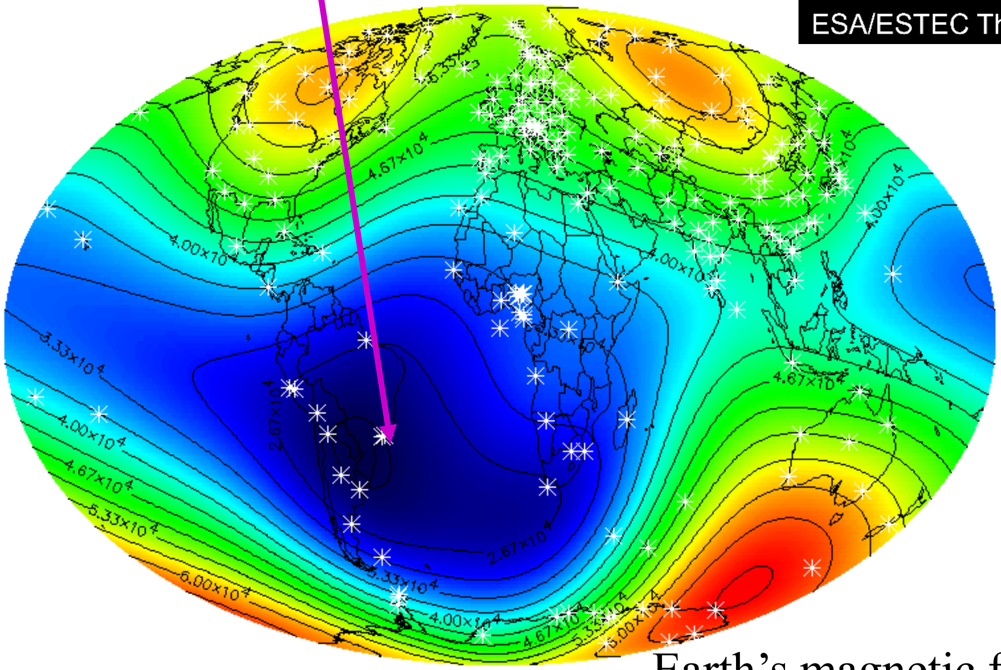
• Microprocessor program corruption.

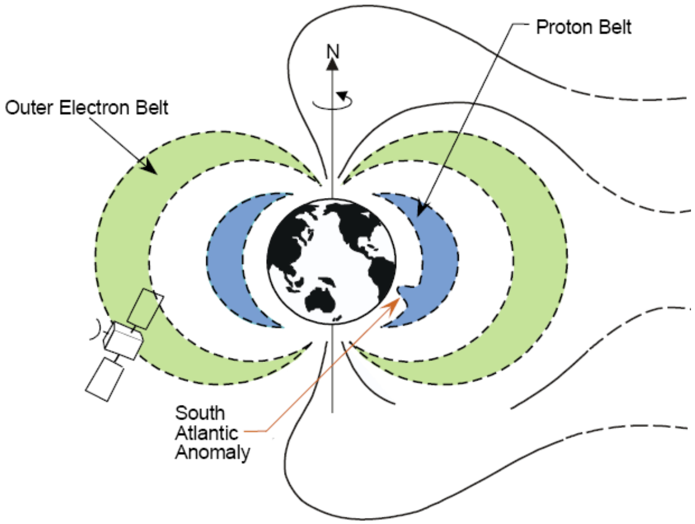
⇒ Calculation errors, freeze (requires a reset), wrong command execution.

Example: Single event memory upsets



South Atlantic Anomaly (SAA)





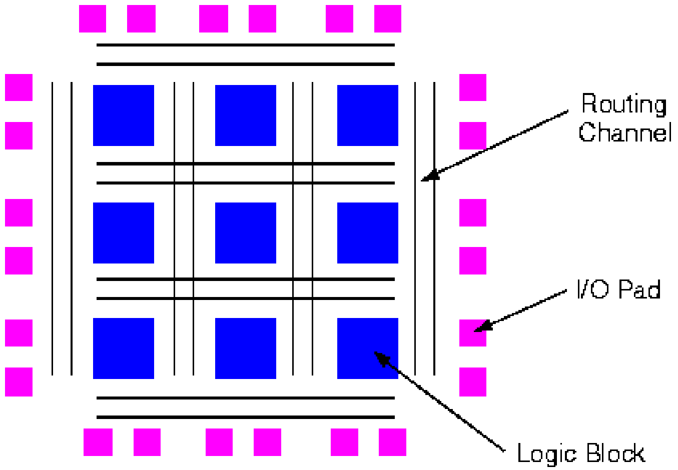
Earth’s magnetic field

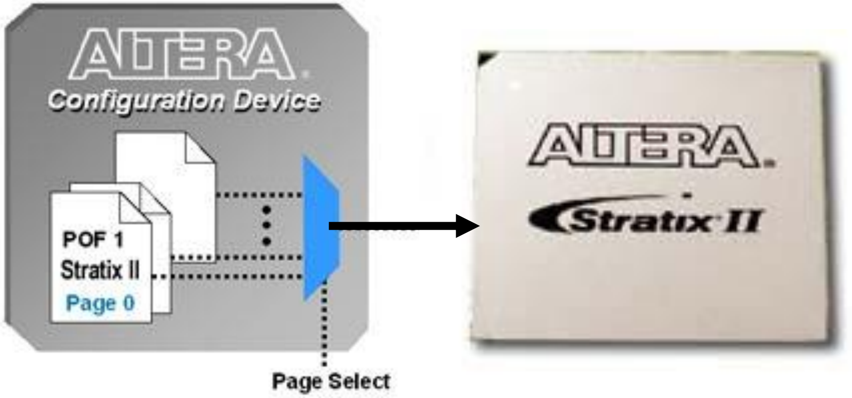
Radiation Countermeasures

- If possible, chose an orbit with a reduced level of radiation.
- Shielding to lower the radiation dose level (using e.g. Al, Cu)
  - Unable to deal with high-energy particles.
- Radiation hardened (rad-hard) components
  - Special manufacturing processes of the electronics, like Silicon-On-Insulator (SOI) technology.
- **System-level error corrections (radiation-hardening by design)**
  - Error detection and correction of memory (parity bits, Hamming code)
  - **Triple Redundancy and Voting (TMR - Triple Module Redundancy)**
    - **Three copies of the same circuit + a voter performing a “majority vote”.**
    - E.g. three separate microprocessors, all doing the same computations.
  - Watchdog timer to avoid processor crash; resets the system automatically if an error is detected.
- Turn off supply voltage before entering a part of the orbit where high radiation is expected
  - Reduce the effect of the ionization.

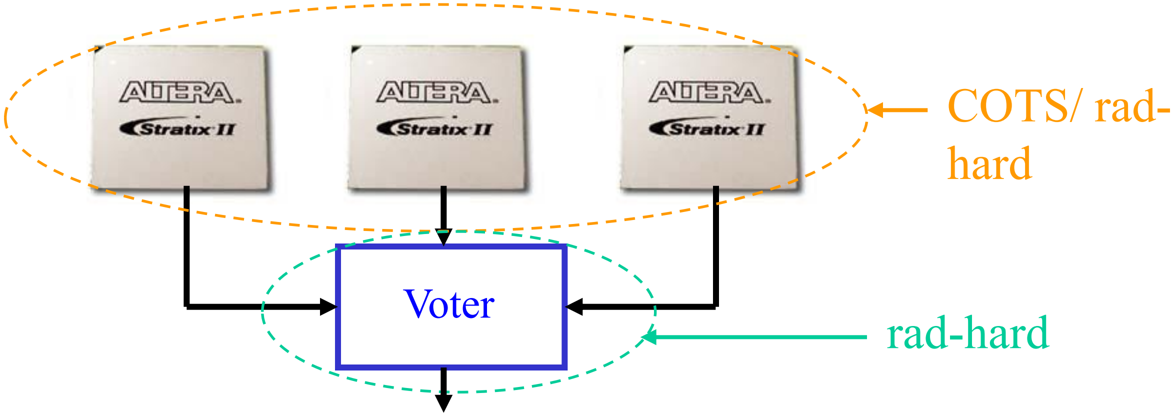
Example: FPGAs and TMR

- FPGA = Field Programmable Gate Array





Fault-tolerant (TMR) FPGA design:



Voter

COTS/ rad-hard

rad-hard