UNIVERSITÀ
DEGLI STUDI
DI PADOVA

An Example of Hardware/Software Partitioning in SDR Systems: the Hardware

Daniele Vogrig, Assistant Professor, DEI Brixen, July 1 2010
vogrig@dei.unipd.it

Preliminary Remark

This is the second of a series of two talks illustrating an example of hardware/ software partitioning.

- This first talk, presented yesterday by Carlo Fantozzi, have introduced the subject, the DVB-T2 system and others aspects
- In this talk, we will present some Hardware aspects with focus on the complete design flow

Outline

- DVB-T2 architecture
- Hardware Scenario
- Hardware basics
- The Lyrtech Board
- Design Flow
- Results
- Future Works

DVB-T2

The DVB-T2 standard

- First version published in September 2009
- Addresses the needs of countries after they have completed Analogue Switch-Off (ASO).
- Leverages on advances in modulation and coding technology to use valuable UHF/VHF spectrum freed by ASO in the most efficient way
- Still uses OFDM. Increased number of modes for extra flexibility.
- Improvement: rotated constellations provide additional robustness
- New FEC encoding: LDPC (Low Density Parity Check) coding combined with BCH (Bose-Chaudhuri-Hocquengham) coding offers excellent performance in the presence of high noise levels and interference
- Improvement: a transmitter diversity method (Alamouti coding) increases coverage in small-scale single-frequency networks

DVB-T2 Transmitter

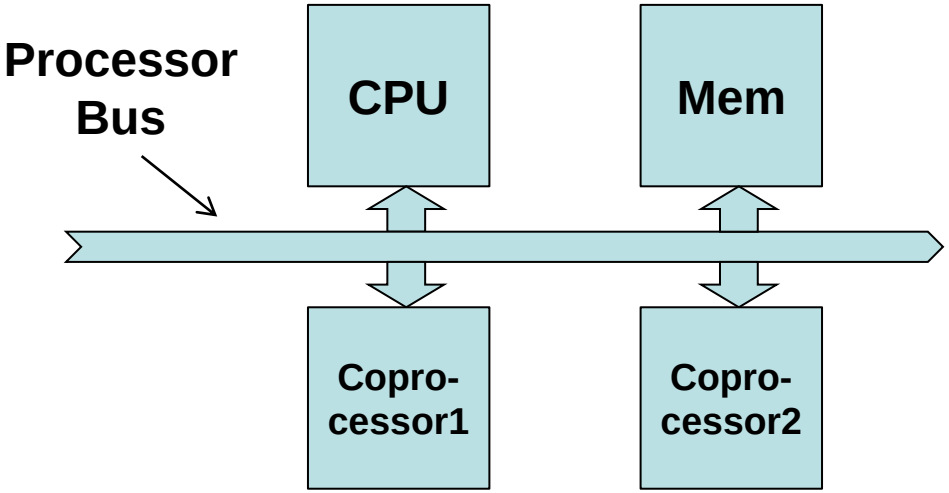
- CRC generation
- PRBS scrambling
- BCH encoding
- **LDPC encoding**
- Bit interleaving
- Gray mapping
- Constellation rot.
- Alamouti coding
- **Inverse FFT**

- DVB-T2 transmitter requires a lot of computations.
- We can realize entire system using different solutions.

~~Use commercial microprocessors (one or more)~~

Use commercial microprocessors for basic computation AND one or more dedicated co-processors

System Architecture



- Co-processor
  - Hardware dedicated only to specific function
  - Optimized (not generic)
  - Reduce CPU load

Application Specific Integrated Circuit (ASIC)

Using which tecnology?

Hardware Scenario

Tecnology Overview

Catalogue components

µC, µP, DSP,
SRAM, DRAM,
SSI, …

Programmable components

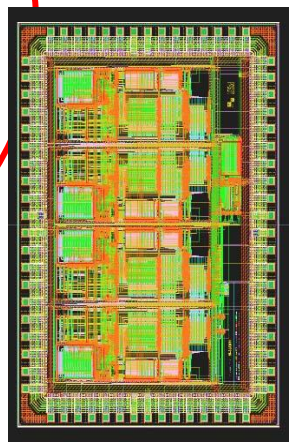
PROM, PAL,
PLA

FPGA, CPLD

Gate Arrays

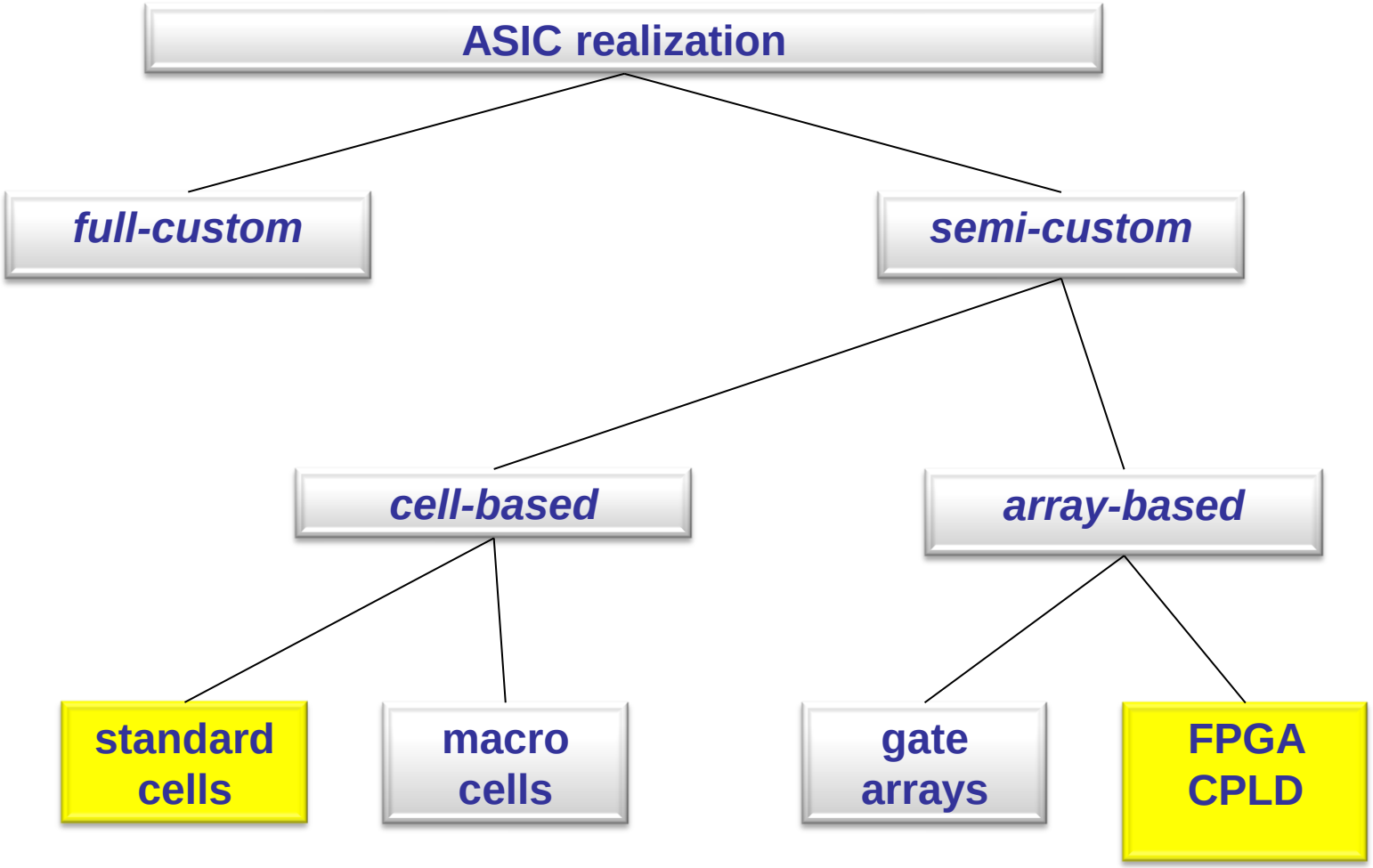
Standard Cells

Full-custom



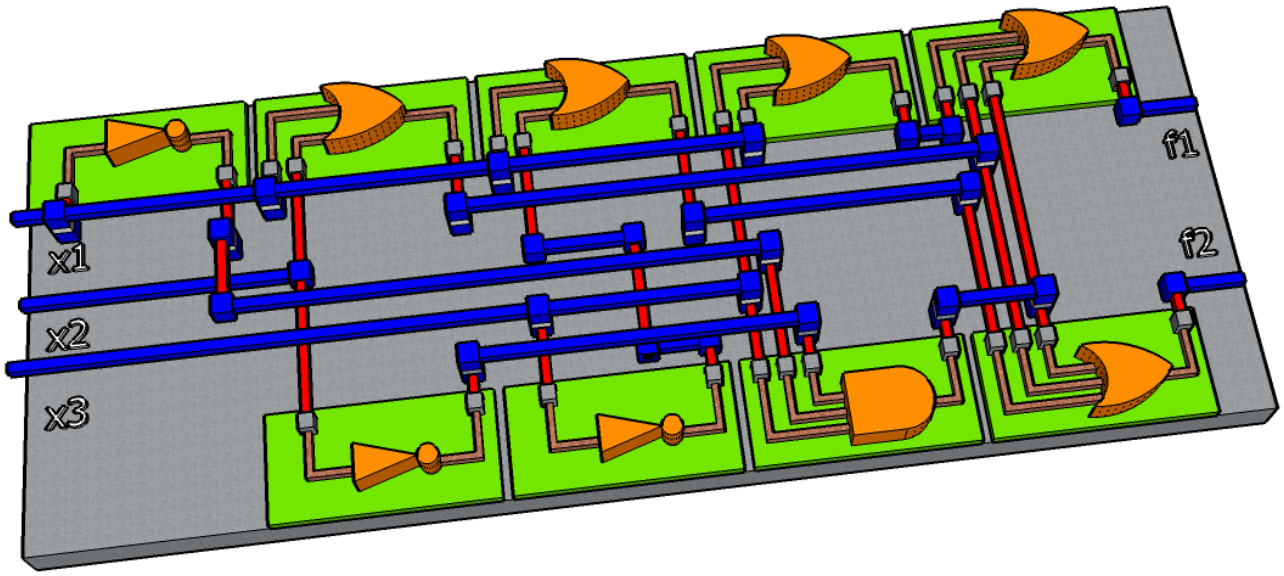
ASIC (Application Specific Integrated Circuit)

ASIC


ASIC realization
full-custom
semi-custom
cell-based
array-based
standard cells
macro cells
gate arrays
FPGA CPLD

Standard-cell

- Use a library of pre-designed cells (completely, up to layout level)
- Cells are place on multiple rows
- The interconnections are realized in the channels between the rows of cells
- The complete layout is sent to the foundry for the manufacture



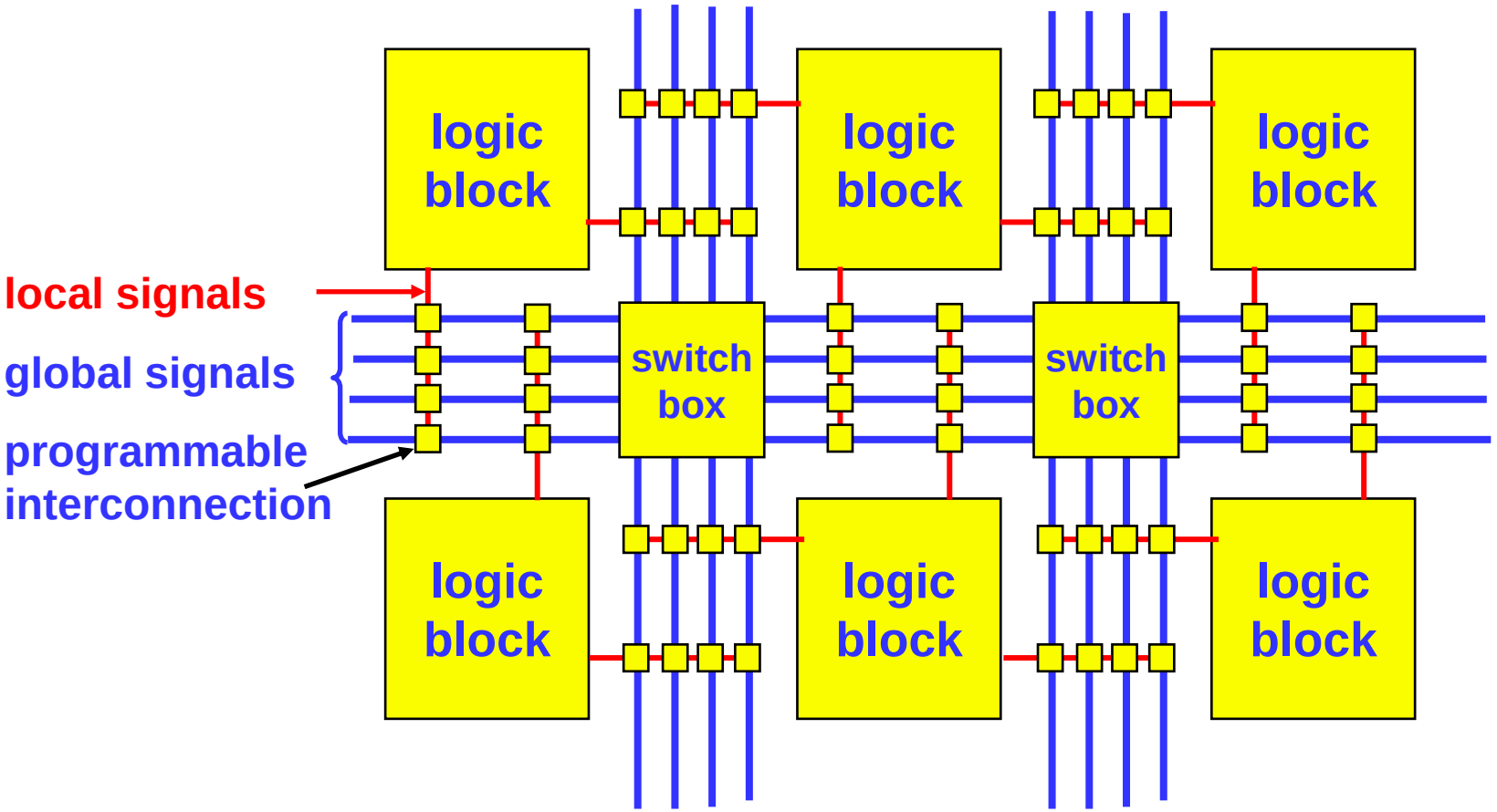
Standard-cell: pro and con

- Optimized performance on each individual cell
  - It has been designed in order to maximize performance
- Full exploitation of resources
  - We implement only the necessary cells
- CAD tools for the design
  - Rapid development
- Library design is expensive
  - Someone has to realize and to update the library (usually the foundry makes this, and you pay for it)
- The complete manufacturing process is needed
  - Starting from the plain wafer, such as full-custom

FPGA: basic structure

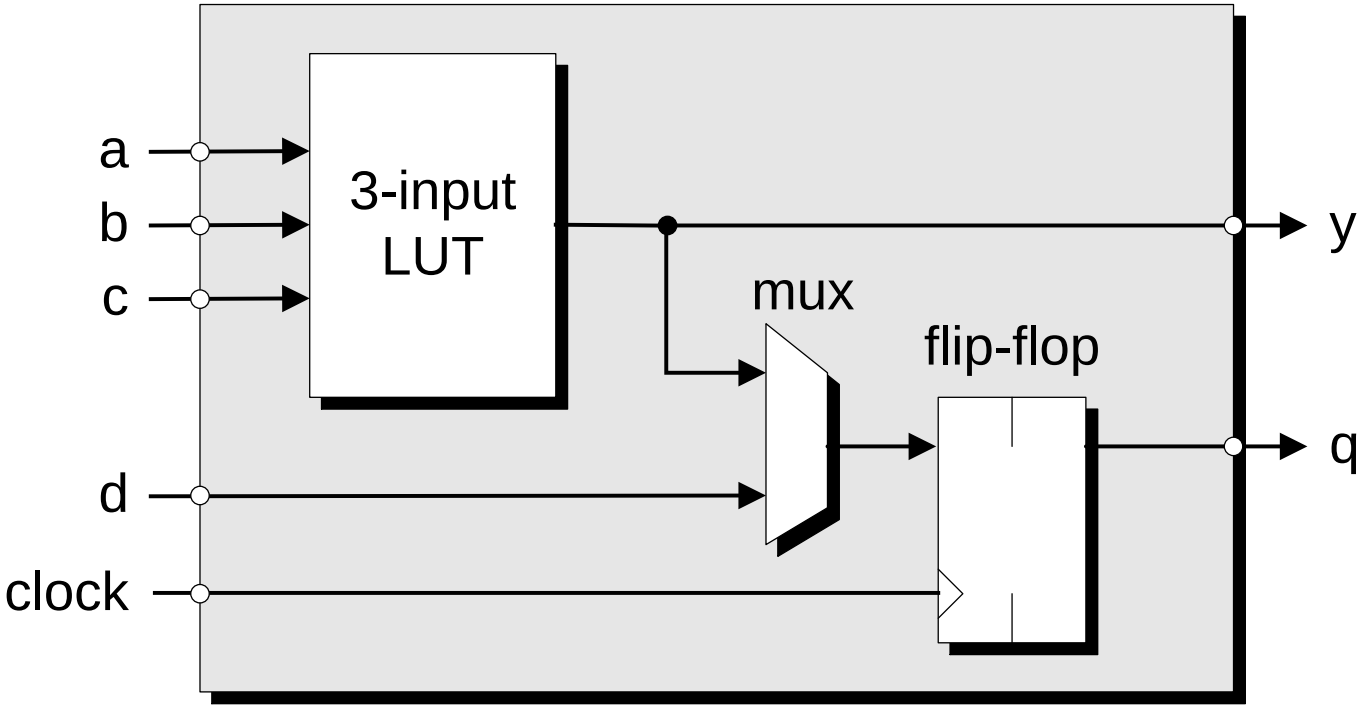
Logic block: it is a pure combinational logic module (with one register or more) and its function is fully programmable

***Switch box*: allow to chose which signal are connected**



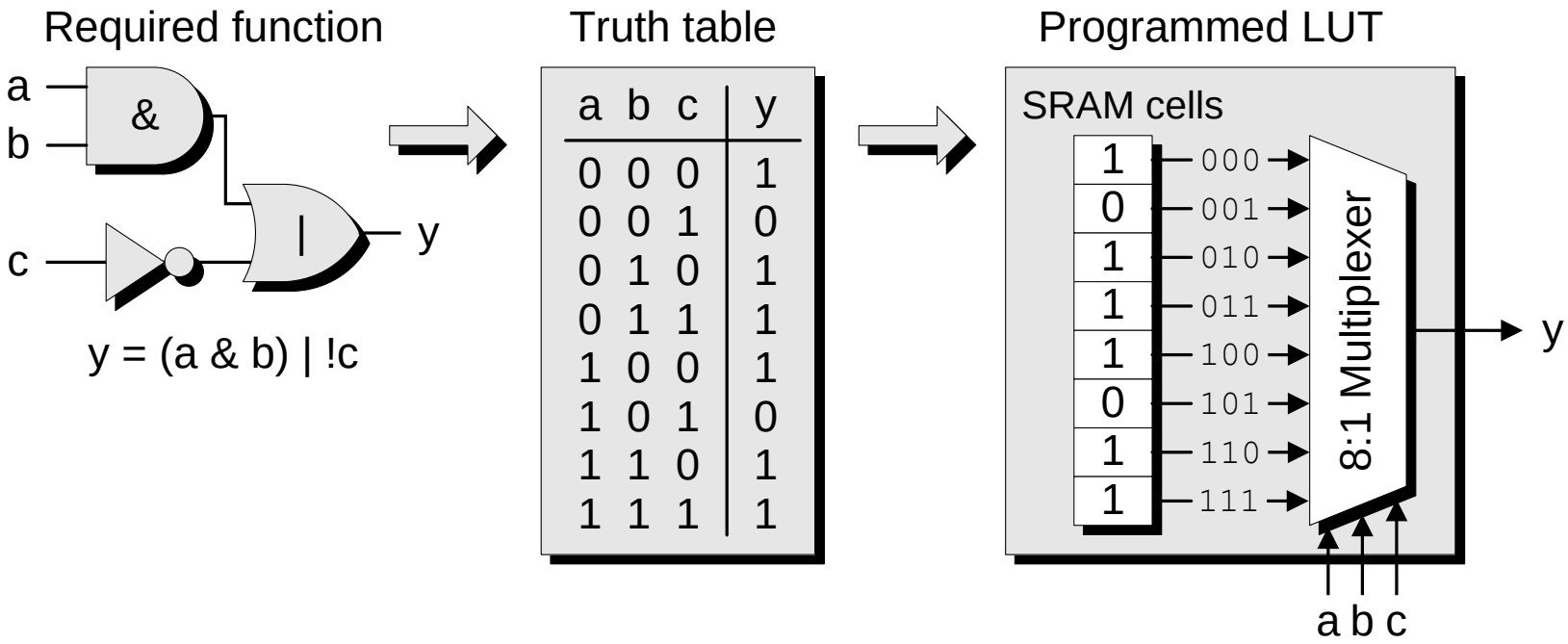
FPGA: generic logic block

- The generic logic block of the FPGA is called *Slice*
- Programmable look-up table (LUT) : it could be configured to represent any 3-input logical function
- It's possible to have a registered output
- Aux input increase the flexibility

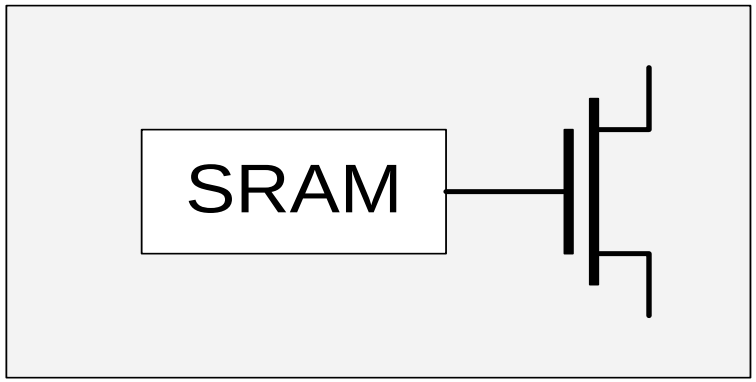


FPGA: programming

- **LUT**



- **Interconnection**

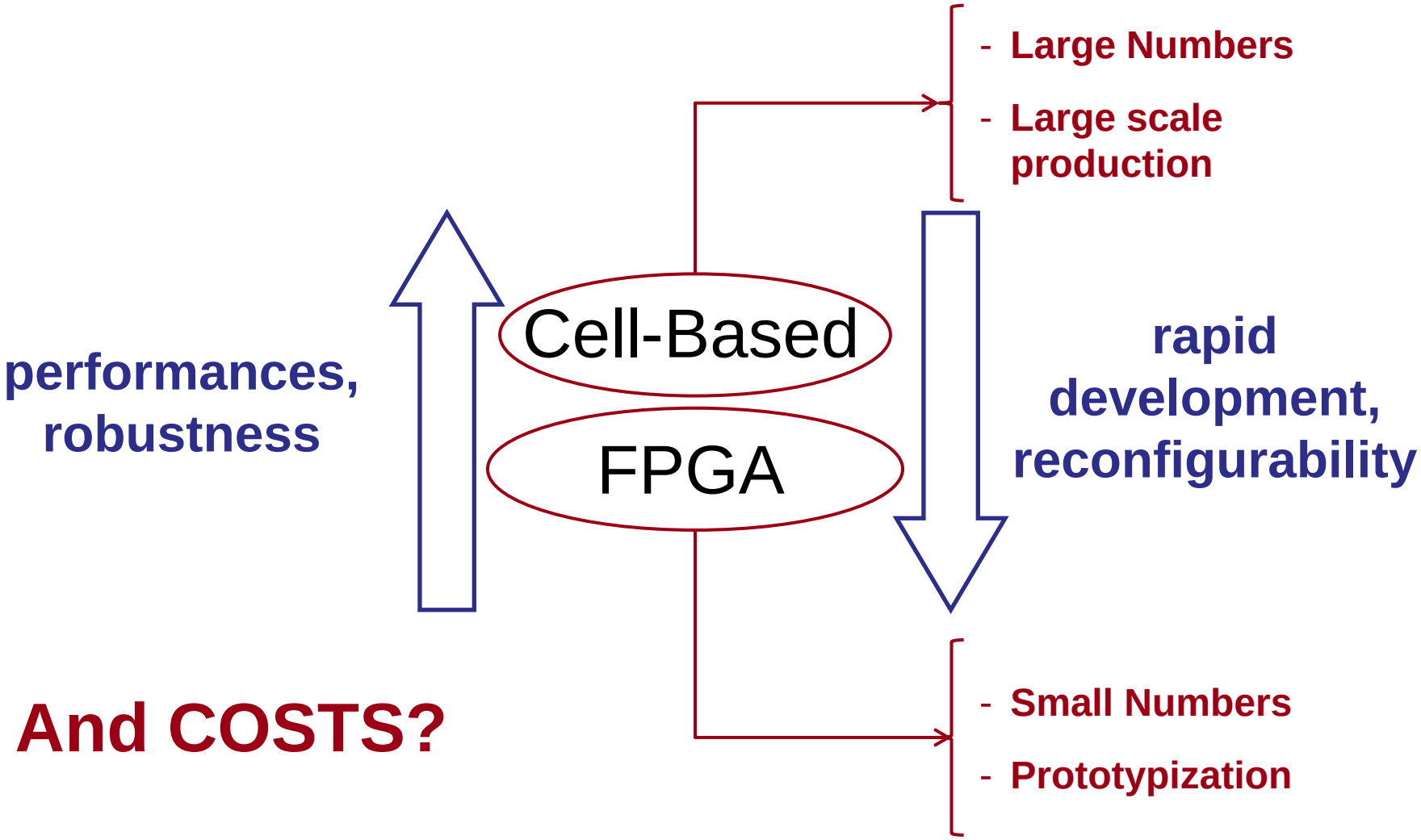


0 = open
1 = close

FPGA: pro and con

- You can buy the finished component and it is programmed directly on the field, without additional fabrication steps
- CAD tools like in standard-cells
- Programming phase takes only some seconds
- The component can be re-programmed (except in some cases)
- You have unneeded cells and interconnections
- Reduced performance (compared to the potential of technology)
- Non-competitive const for large numbers (20,000 ÷ 50,000)

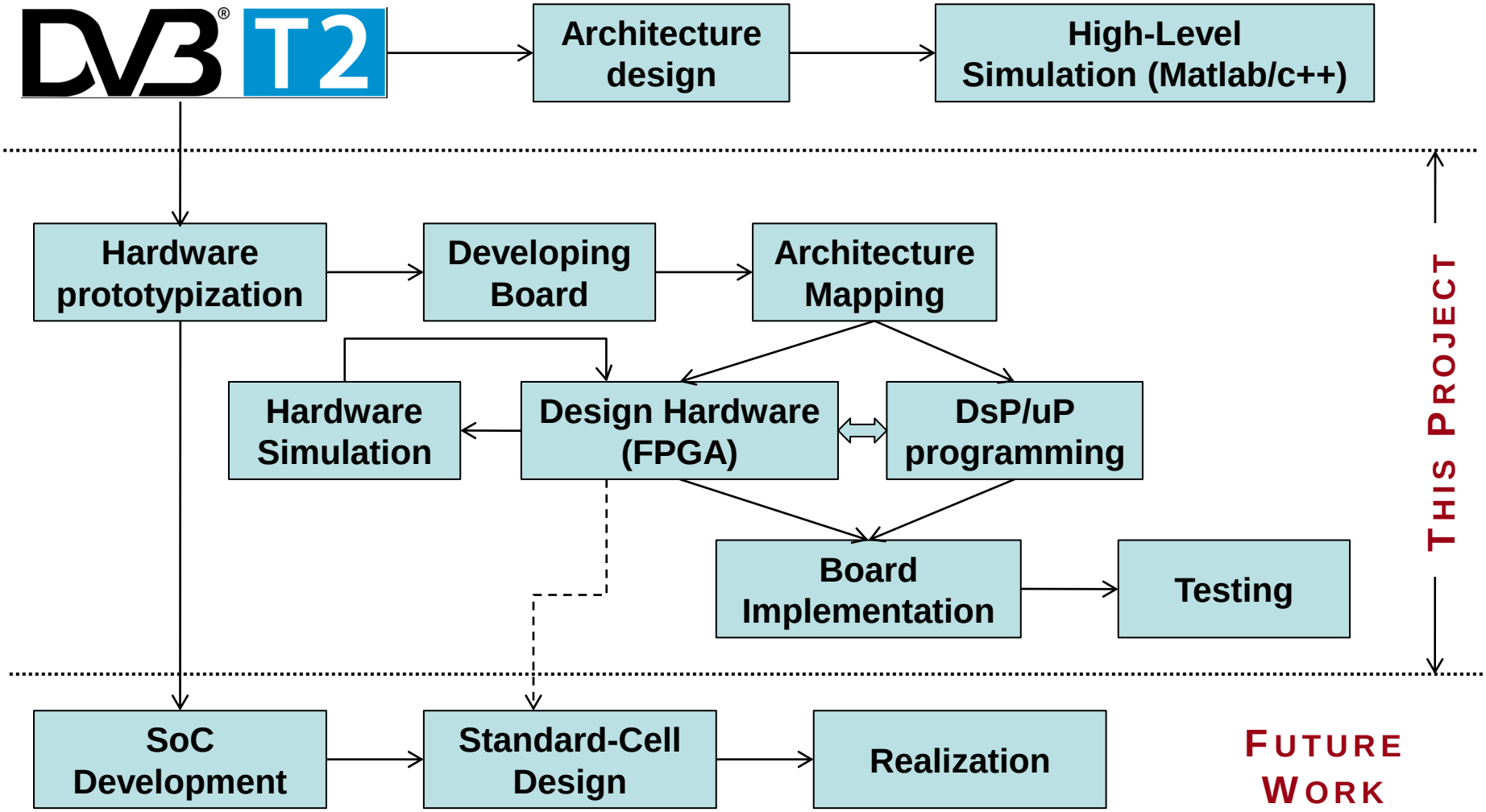
Which tecnology?



And COSTS?

Hardware Prototyping

Our project


DVB T2
Architecture design
High-Level Simulation (Matlab/c++)
Hardware prototypization
Developing Board
Architecture Mapping
Hardware Simulation
Design Hardware (FPGA)
DsP/uP programming
Board Implementation
Testing
SoC Development
Standard-Cell Design
Realization
THIS PROJECT
FUTURE WORK

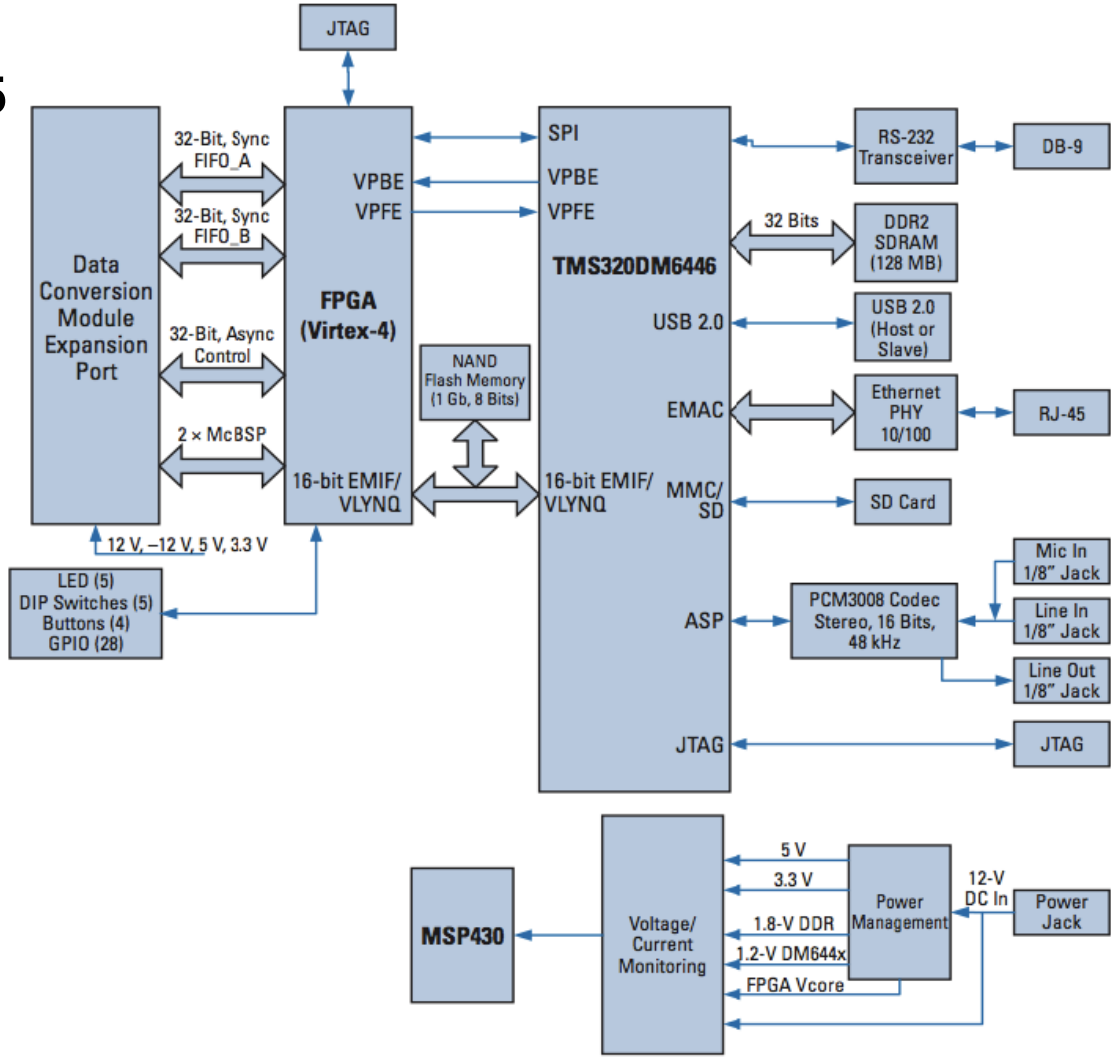
The Lyrtech Board

Designed for
- public safety applications like TETRA and APCO band communications
- vehicular systems
- transponders
- RFID readers
- WiMAX and Wi-Fi customer-premises equipment (CPE)
- broadband data systems
- femto and pico base stations

Three distinct modules:
- **RF Module**
- **Data Conversion Module**
- **Digital Processing Module**

The Lyrtech Board: Digital processing module

- **GPP+DSP: TI "DaVinci"**
- **FPGA: Xilinx Virtex-4 SX35**
- MCU: TI MSP430 MCU
- 128-MB DDR2 SDRAM
- 128-MB NAND Flash
- Stereo audio codec
- 10/100-Mbit/s Ethernet
- JTAG probing access
- HMI: LEDs, buttons, dip switches



Data-Rate

Worst case ➔Data rate: ≅ 61 megabits per second

TI "Da Vinci"

- **GPP**: ARM926EJ-S @ 297 MHz
  - Neither the LDPC encoder not the iFFT can be implemented on the GPP
- **DSP**: T64x+ core @ 594 MHz
  - Theoretically it can realize both functions
  - But we need to use proprietary software developed by TI ➔ **NO GOOD!**

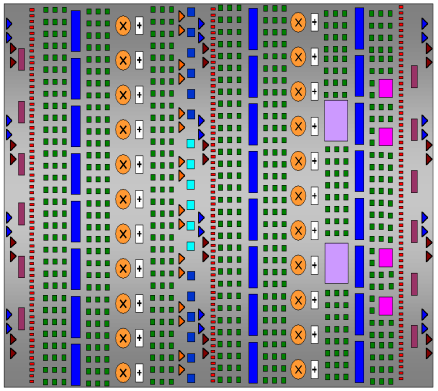
Xilinx Virtex 4 FPGA

- Up to 450 MHz clock
- Various implementation of LDPC encoders and iFFT in literature

We decide to implement both LDPC encoder and iFFT on board's FPGA

Xilinx Virtex-4 XC4SX35

- Consist in an array (96x40) of Configurable Logic Blocks (CLBs)
- Each CLB contains 4 *slices* (Tot.15,360)
- Each slice contains:
  - 2 flip-flops (Tot. 30,720)
  - 2 programmabe 4-in LUTs (Tot. 30,720)
- Half of LUTs can be used as distributed RAM or shift registers (Max of 240 kilobits)
- Contains 192 XtremeDSP® embedded blocks
- Contains 192 Block RAM (Tot. 3,456 kb)
- Max of 448 I/O Pins



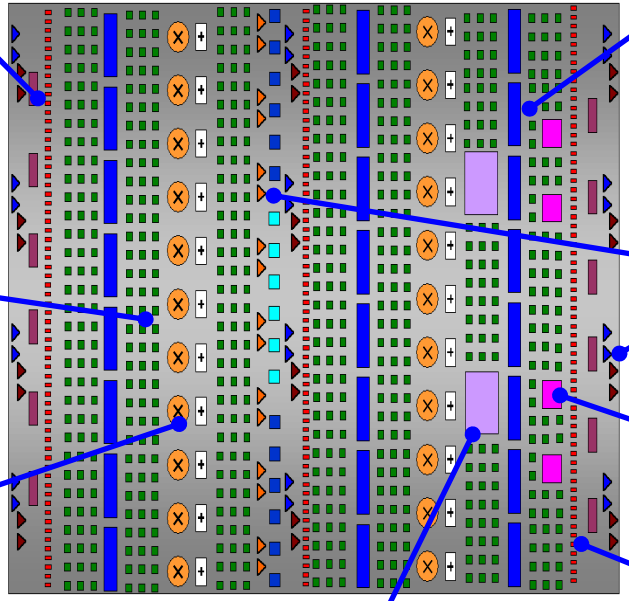
Virtex-4 Family

RocketIO™ Multi-Gigabit Transceivers
622 Mbps–10.3 Gbps

Advanced CLBs
200K Logic Cells

XtremeDSP™ Technology Slices
256 18x18 GMACs

PowerPC™ 405 with APU Interface
450 MHz, 680 DMIPS



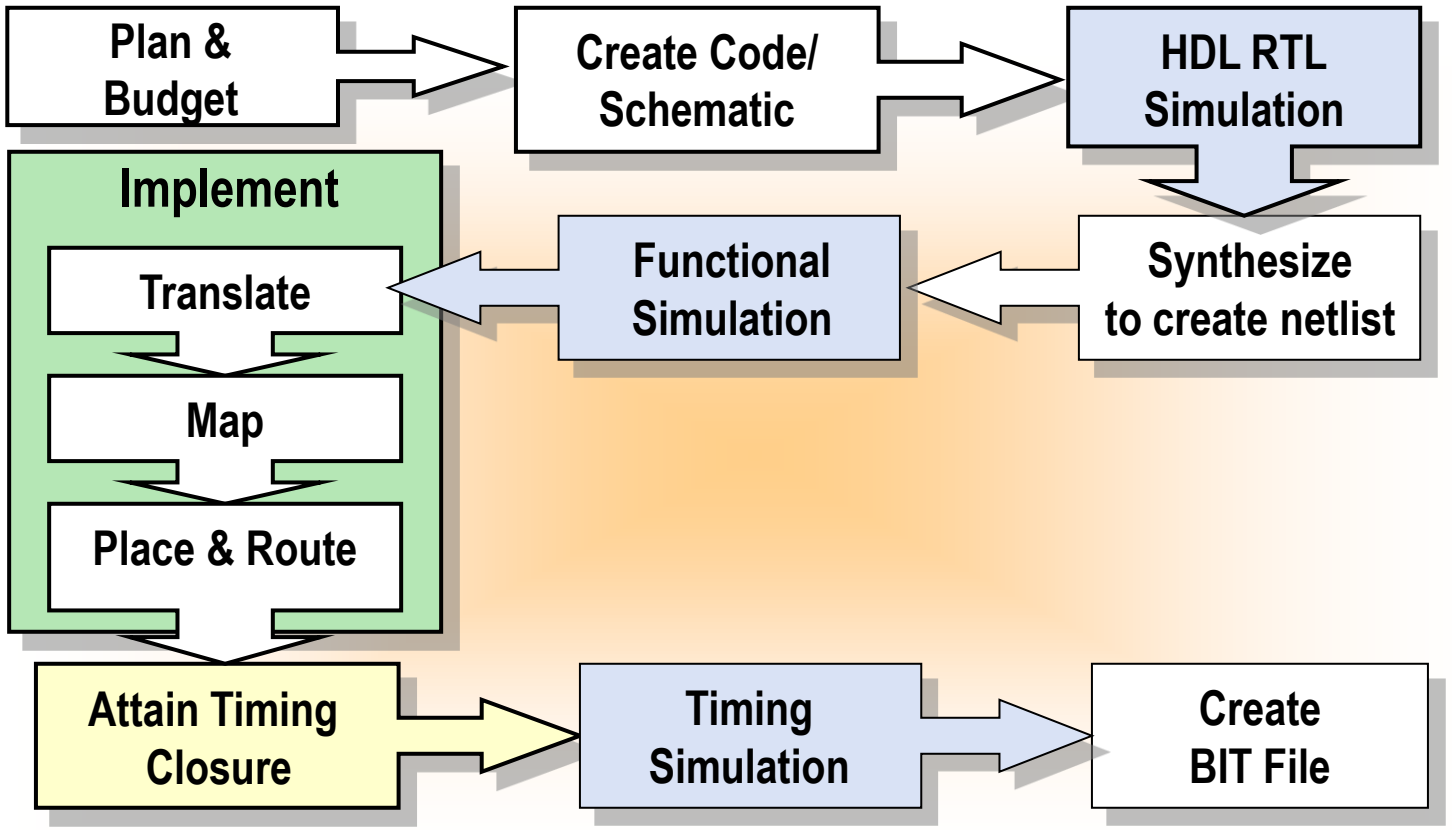
Smart RAM
New block RAM/FIFO

Xesium Clocking Technology
500 MHz

Tri-Mode Ethernet MAC
10/100/1000 Mbps

1 Gbps SelectIO™
ChipSync™ Source synch, XCITE Active Termination

Xilinx Design Flow



LDPC encoder

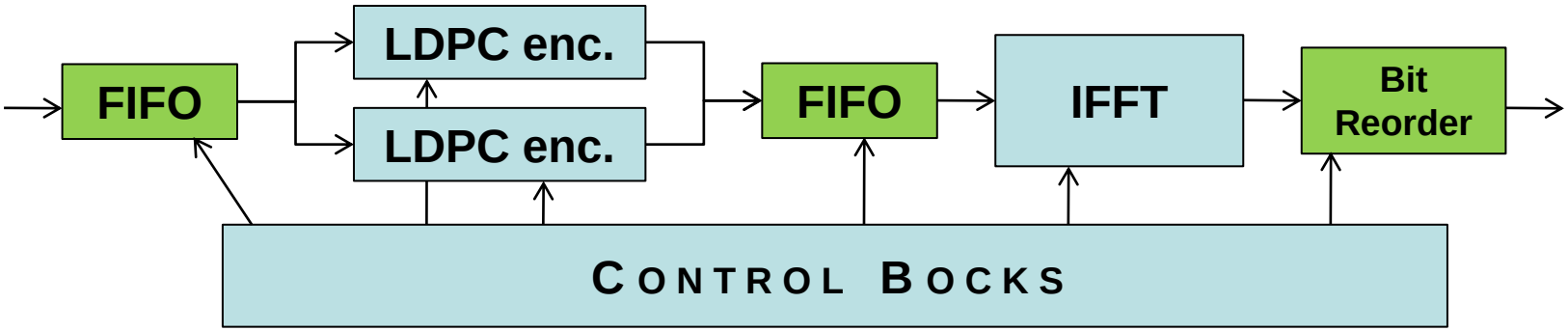
- Classical structure proposed by Richardson and Urbanke*
- DVB-T2 LDPC tables are memorized in FPGA block RAMs
- Iterative core has been design and tested successfully
  - Design developed using VHDL language
  - Using Xilinx simulation tools and comparing results with high-level description
- Performance:
  - Max clock frequency: 110 MHz
  - Use 2% of CLBs and 15% of BlockRAMs
  - Encoding Time: 2,7 ms (@WC)
- In order to achieve the required data-rate, we have decided to implement 2 encoders in a ping-pong structure

[*] T.J. Richardson; R.L. Urbanke “Efficient encoding of low-density parity-check codes,” IEEE Transactions on Information Theory, vol. 47, n.2, pp. 638–656, Feb. 2001.

IFFT

- Must be run-time reconfigurable (up to 32K points)
- We adopted a pipeline architecture based on radix-2 butterflies
- The input data width is 8 bit for real + 8 bit for imag.
  - This width can be easily changed in VHDL code
- Internal data width is 10 + 10 bit
  - We also implement internal rounding for overflow control
- Performance:
  - We use the LDPC encoder clock: 110 MHz
  - The pipelined structure can elaborate 1 symbol (8+8 bits) each clock period
  - Maximum Latency: 597 μs
  - Use 20% of CLBs, 56 BlockRAMs (30%) and 28 ExtremeDSP (15%)

Overall system

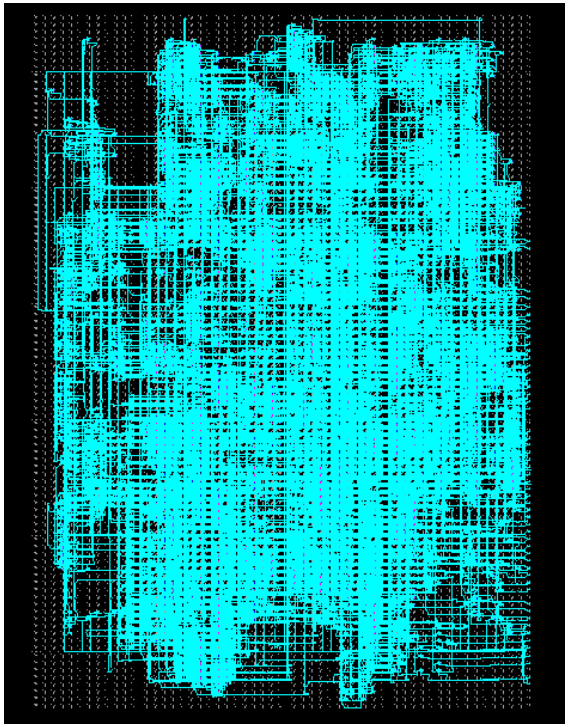


- Some FIFOs are added in order to obtain the correct data-flow
- At the output of IFFT, an bit reorder module is added
- We have designed also the needed control blocks for the system
- The entire system is described using VHDL language
- When possible, we have maintained a generic VHDL structure, in order to maximize the portability of description

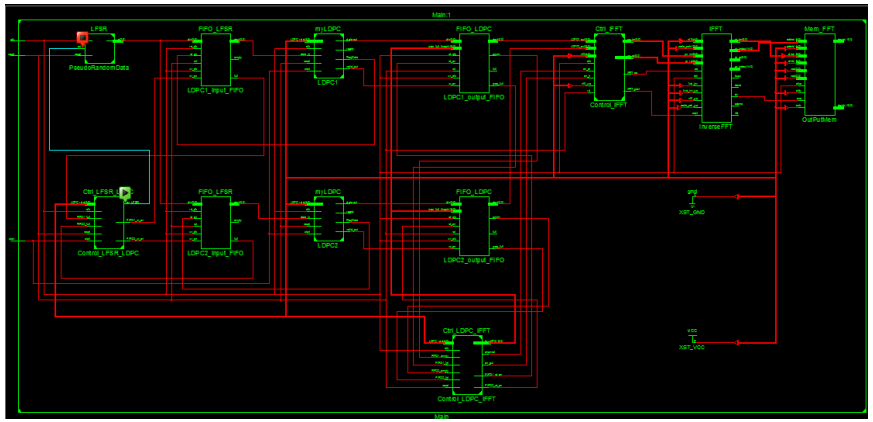
Implementation

- Implemented on the XC4SX35 FPGA the entire system uses about 65% of BlockRAMs and less then 30% of the other structures.
- A simulation of the hardware realization has been done and compared with high level model of the system
- We have planned a hardware testing (to be done)
  - We need additional structure to support bi-directional communication between FPGA and TI “DaVinci”

Implementation view in ISE Xilinx Design Tool



Schematic view in ISE Xilinx Design Tool



Future Work

- Complete the system
  - Add communications block for data-transfer between “Da Vinci” and FPGA
  - Add others DVB-T2 block (i.e. BCH encoder, etc)
  - Define a full testing strategy
- Realize using standard-cell technology
  - Design Porting
  - Re-design of dedicated block (i.e. multipliers)
  - Microprocessor realization

Thansk!

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

An Example of Hardware/Software Partitioning in SDR Systems: the Hardware

Daniele Vogrig, Assistant Professor, DEI Brixen, July 1 2010