



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

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

Carlo Fantozzi, Assistant Professor, DEI

Brixen, June 30 2010



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Preliminary Remark

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

- This first talk will concentrate on telecom/protocol/software aspects
- Hardware aspects will be dealt with tomorrow by Daniele Vogrig during the second talk





UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Summary of the Presentation

- The concept of Software Defined Radio (SDR)
- The DVB-T2 standard
- The DVB-T2 transmitter
- The SDR hardware we chose:
Lyrtech Small Form Factor Software Defined Radio
(SFF SDR) board
- Details about software
 - Operating systems
 - Libraries
 - Communication among components



SDR

Software Defined Radio



SDR = **S**oftware **D**efined **R**adio (J. Mitola, 1991)

SBR = **S**oftware **B**ased Radio

SR = **S**oftware **R**adio

...

Different shades in different definitions.

Broad aim: “to shift from employing a traditional hardware-focused, application specific approach to radio implementation to using a **software application** to perform the radio tasks on a **computing platform**” (*Stephen M. Blust, Cingular Wireless*)





SDR devices are

- **multi-mode**
- **multi-band**
- **multi-functional...**

They can be changed, enhanced, and upgraded on-the-fly through hardware reconfiguration and/or software updates.

Therefore, **SDR** devices include **both software and hardware that can be dynamically reconfigured** to enable communication between a wide variety of changing communications standards, protocols, and radio links.



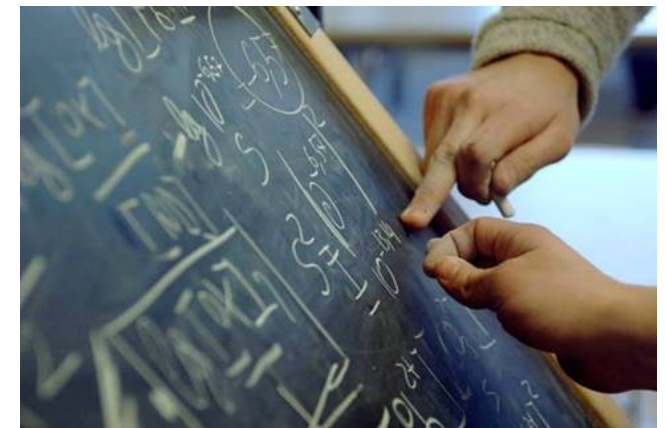


Pros

- Much more flexible
- Easier to program
- Easier to fix
- Easier to change/enhance
- Same hardware in multiple apps

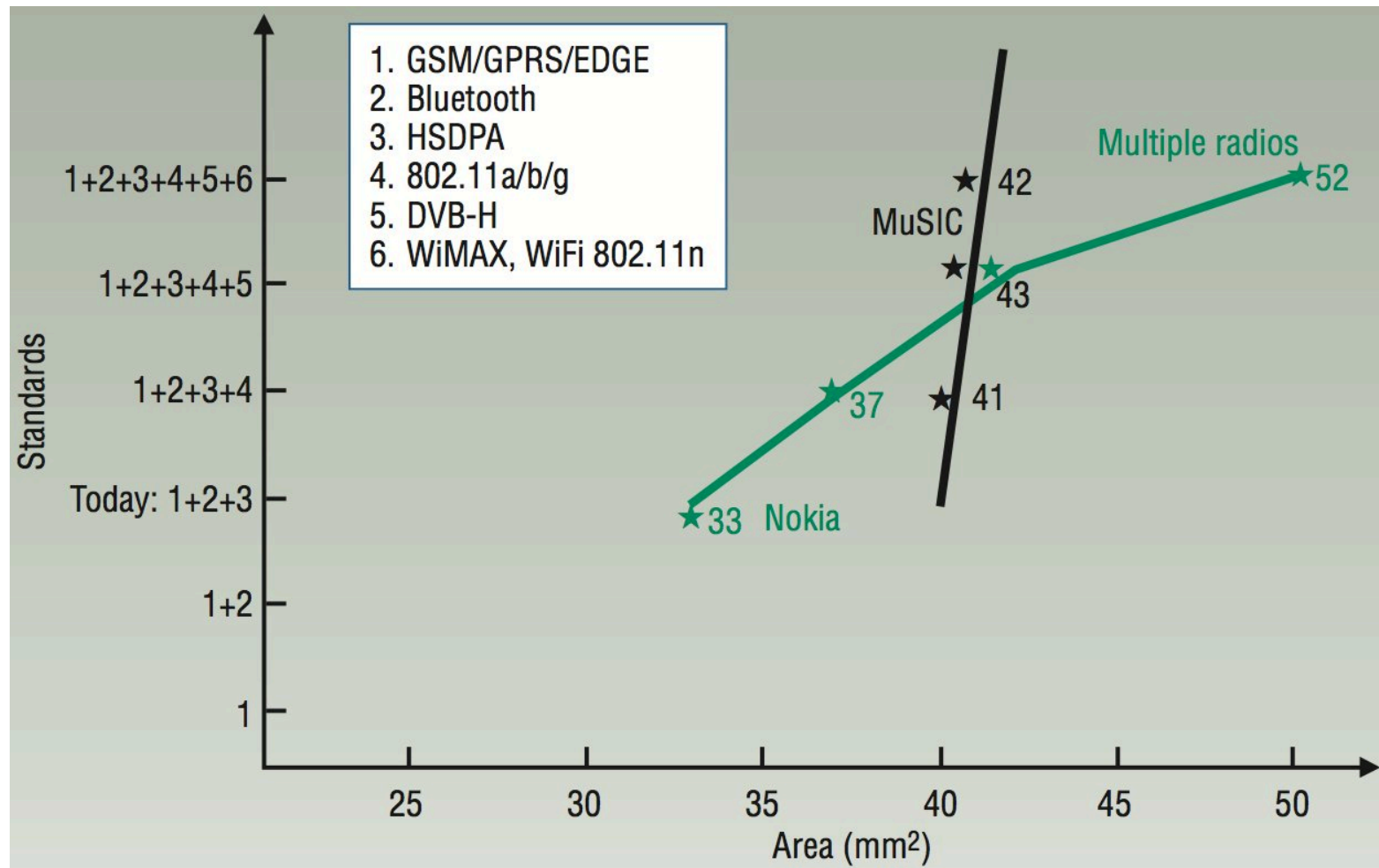
Cons: less “efficient”

- Higher power consumption
- More chip area (but...)





Cell phones: area consumption of multiple radios vs. SDR radios @65 nm





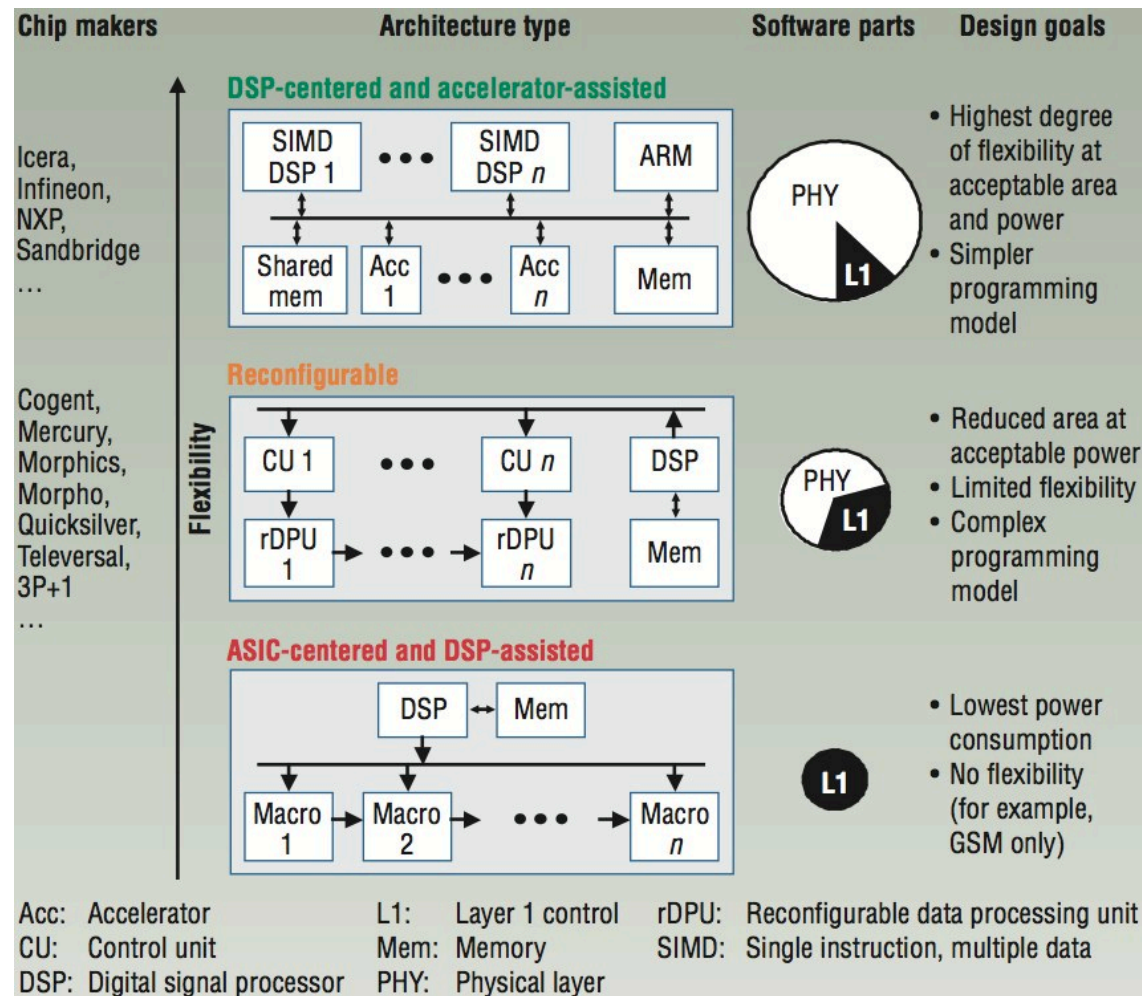
Compromise: architecture made of
heterogeneous cores.

- **FPGA**: (Field Programmable Gate Array):
baseband modulation/demodulation, filtering,
channel coding and decoding
- **DSP** (Digital Signal Processor):
source (maybe channel) coding/decoding
- **GPP** (General Purpose Processor):
user applications and communication protocols





Example: baseband architectures for multistandard mobile phones





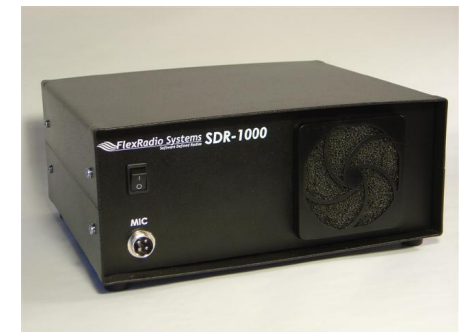
UNIVERSITÀ
DEGLI STUDI
DI PADOVA

SDR: Vendors

Many different vendors of SDR platforms

- GE Fanuc
- Innovative Integration
- Intelligent Software Radio Technologies (→IDP100)
- Lyrtech (→SFF SDR)
- Pentek
- Spectrum Signal Processing (→SDR-4000)
- Sundance Multiprocessor Technology
- USRP...

We consider the implementation of a DVB-T2 transmitter on a board with a **Xilinx FPGA** and a Texas Instruments DaVinci chip (**ARM GPP, C64x DSP**)



DVB-T2



DVB-T specifies the **framing structure**, **channel coding** and **modulation** for digital terrestrial television (DTT) broadcasting.

First version published in March 1997.

Now adopted in more than 120 countries (→GE06).

Flexible system that allows networks to be designed for the delivery of a wide range of services.

Nothing is said about video encoding

- MPEG-2 widely popular
- H.264/AVC MPEG-4 increasingly adopted for the delivery of HDTV services





A system for DTT is **required to cope with a variety of noise and bandwidth environments and multipath interference.**

DVB-T uses OFDM (Orthogonal Frequency Division Multiplex) modulation with GI (Guard Interval). Using a large number of sub-carriers, OFDM has the ability to deal with severe channel conditions.

DVB-T has several options to make it flexible:

- 3 modulation options (QPSK, 16QAM, 64QAM)
- 5 different FEC (forward error correction) rates
- 4 Guard Interval options
- Choice of 2k or 8k sub-carriers
- Can operate in 6, 7 or 8 MHz –wide channels (video at 50Hz or 60Hz)

DVB[®]
**Digital Video
Broadcasting**





The New 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





An **increased number of modes** is available in areas such as the number of carriers, guard interval sizes and pilot signals.



- **Overheads can be minimized** for any target transmission channel
- **Higher bit rates can be achieved.**

Example: for an Italian SFN DVB-T profile (64-QAM, 8k, rate 2/3, GI 1/4), the DVB-T2 equivalent (256-QAM, 32k, rate 3/5, GI 1/16), achieves an increase in bit rate from 19.91 Mbit/s to 33.3 Mbit/s (+67%).

	DVB-T	DVB-T2
FEC	Convolutional Coding + Reed Solomon 1/2, 2/3, 3/4, 5/6, 7/8	LDPC + BCH 1/2, 3/5 , 2/3, 3/4, 4/5 , 5/6
Modes	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM, 256QAM
Guard Interval	1/4, 1/8, 1/16, 1/32	1/4, 19/256 , 1/8, 19/128 , 1/16, 1/32, 1/128
FFT size	2k, 8k	1k, 2k, 4k, 8k, 16k, 32k
Scattered Pilots	8% of total	1%, 2%, 4%, 8% of total
Continual Pilots	2.6% of total	0.35% of total



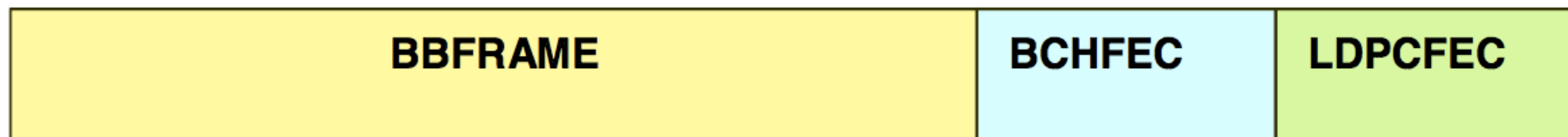
1. A **BCH** code is added to each frame.

This is a well-known cyclic code that can be decoded with simple hardware

2. An **LDPC** code is cascaded to the BCH code.

With LDPC codes, the noise threshold can be set very close to the Shannon limit (for a symmetric memoryless channel).

First developed in 1963, they have been considered impractical until efficient algorithms for encoding (→ iterative encoding) and decoding (→ iterative belief propagation) were discovered in the 90s.





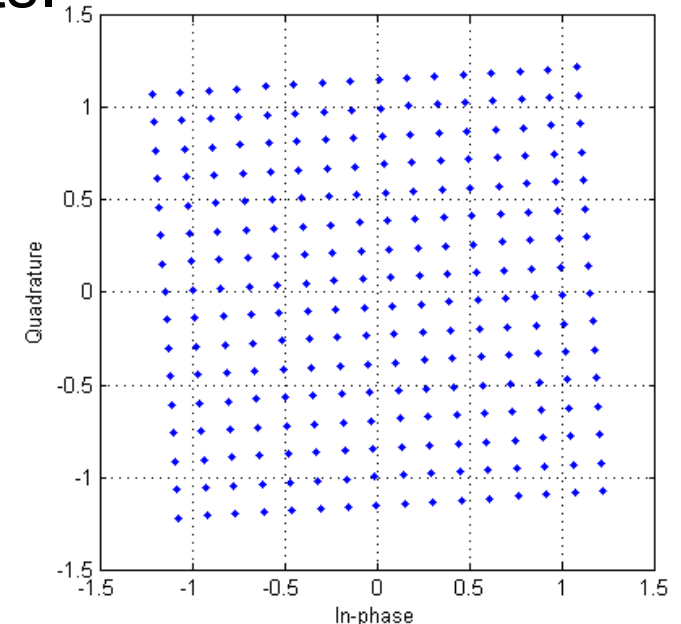
DVB-T2: Constellation Rotation

The normalized cell values of each FEC block are rotated in the complex plane and the imaginary part cyclically delayed by one cell within a FEC block.

This technique **increases the robustness of the DVB-T2 receiver in severe multipath propagation scenarios**, and especially in channels with erasure events.

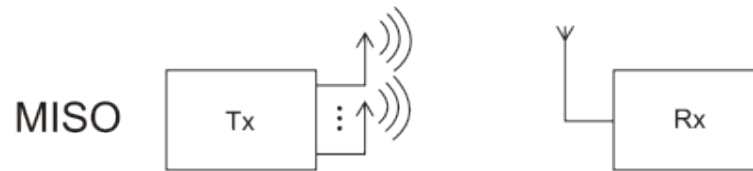


Modulation	QPSK	16-QAM	64-QAM	256-QAM
Φ (degrees)	29,0	16,8	8,6	$\text{atan}(1/16)$



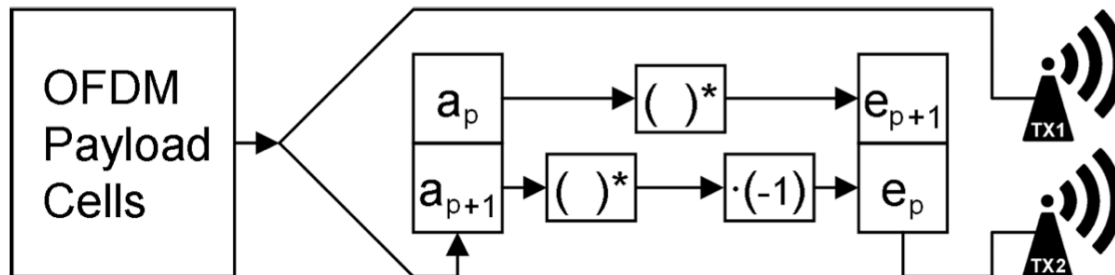


DVB-T2: Transmitter Diversity



An optional MISO encoding can be implemented to reduce fading effects.

Frequency domain coefficients are processed by a **modified Alamouti encoder**, which allows the T2 signal to be split between two groups of transmitters on the same frequency in such a way that the two groups will not interfere with each other.



$$\begin{aligned} e_{m,l,p}(Tx1) &= a_{m,l,p} & e_{m,l,p+1}(Tx1) &= a_{m,l,p+1} \\ e_{m,l,p}(Tx2) &= -a_{m,l,p+1}^* & e_{m,l,p+1}(Tx2) &= a_{m,l,p}^* \end{aligned}$$



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

The DVB-T2 Standard

References

ETSI EN 302 755 V1.1.1

“Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)”

ETSI TS 102 773 V1.1.1

“Modulator Interface (T2-MI) for a second generation digital terrestrial television broadcasting system (DVB-T2)”

DVB BlueBook A122 (draft EN302 755 V1.2.1, rev. 9)

“Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)”

DVB bluebook A133

“Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2)”

DVB BlueBook A150

“Structure and modulation of optional transmitter signatures (T2-TX-SIG) for use with the DVB-T2”

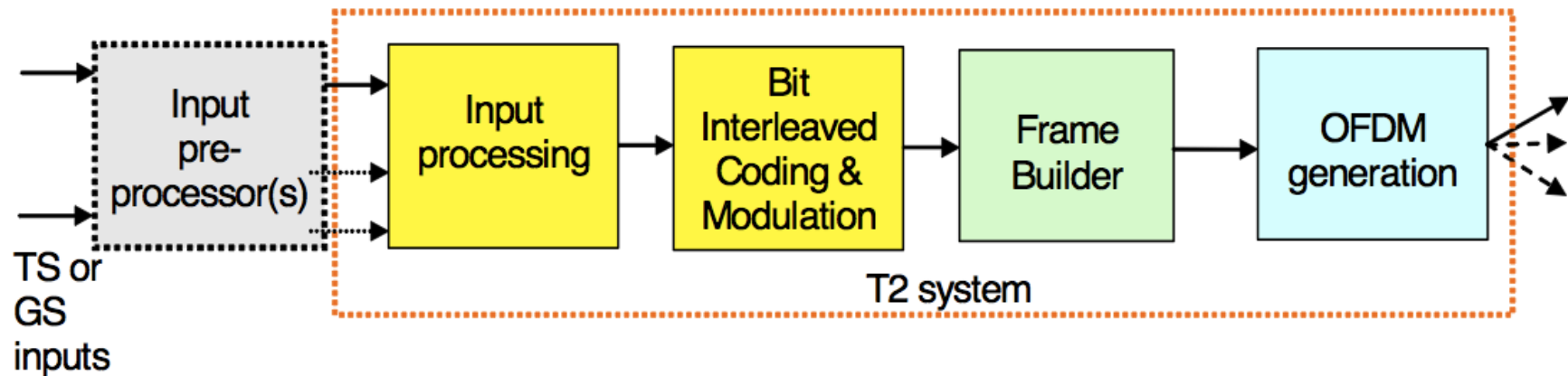
Available at <http://www.dvb.org/technology/standards/> .



**628
pages**



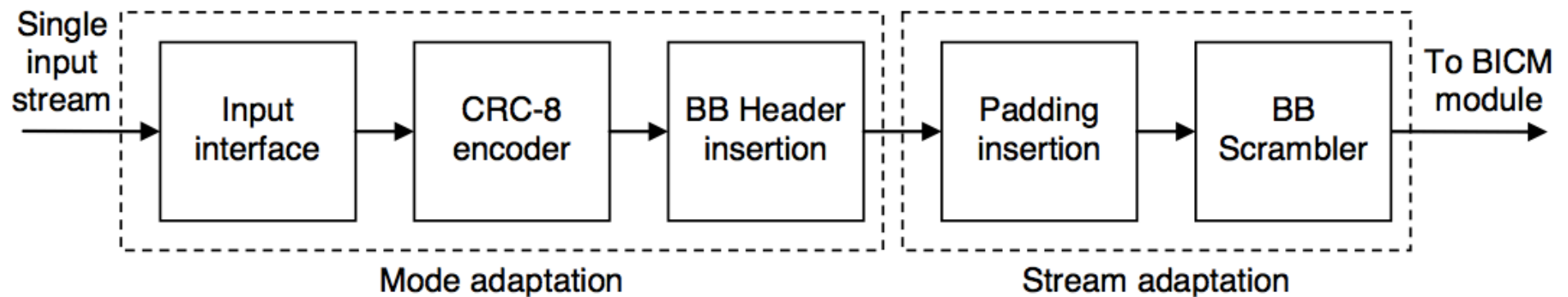
The system input(s) may be one or more Transport Stream(s) and/or one or more Generic Stream(s). The Input Pre-Processor may include a splitter or de-multiplexer for Transport Streams (TS) for separating the services into the **T2 system inputs, which are one or more logical data streams**. These are carried in individual Physical Layer Pipes (PLPs).

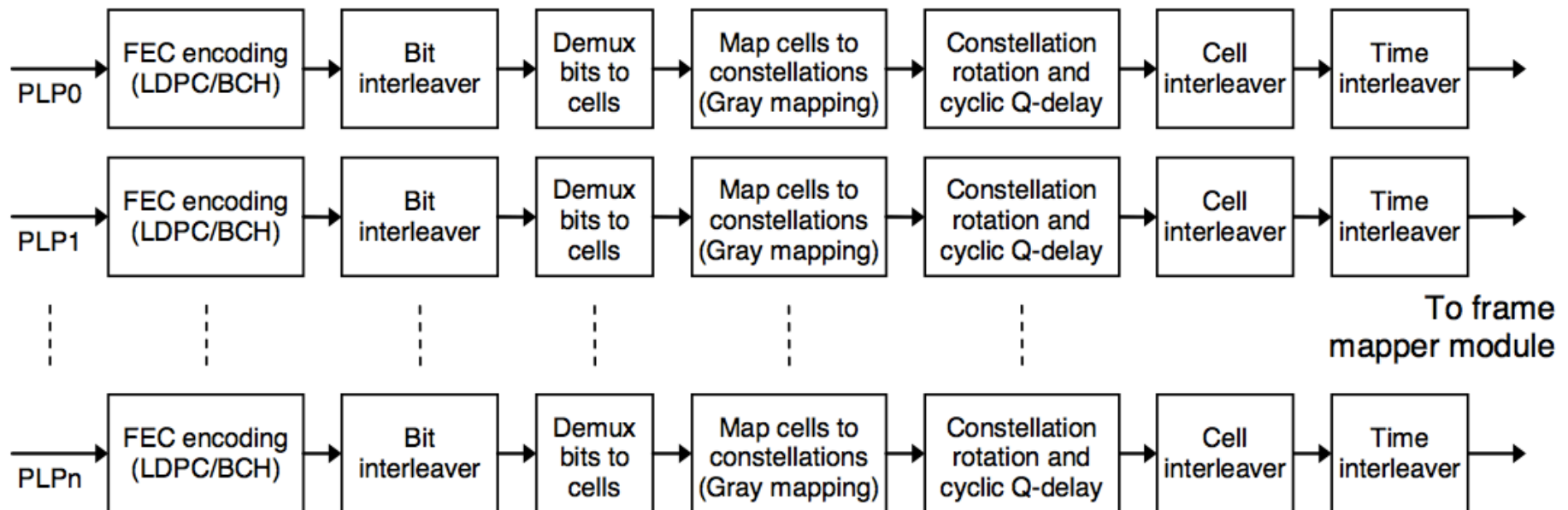


The maximum input rate for any TS, after deletion of null packets when applicable, is **more than 50 Mbit/s** (in an 8 MHz channel).



Input processing module for a single Physical Layer Pipe (PLP)

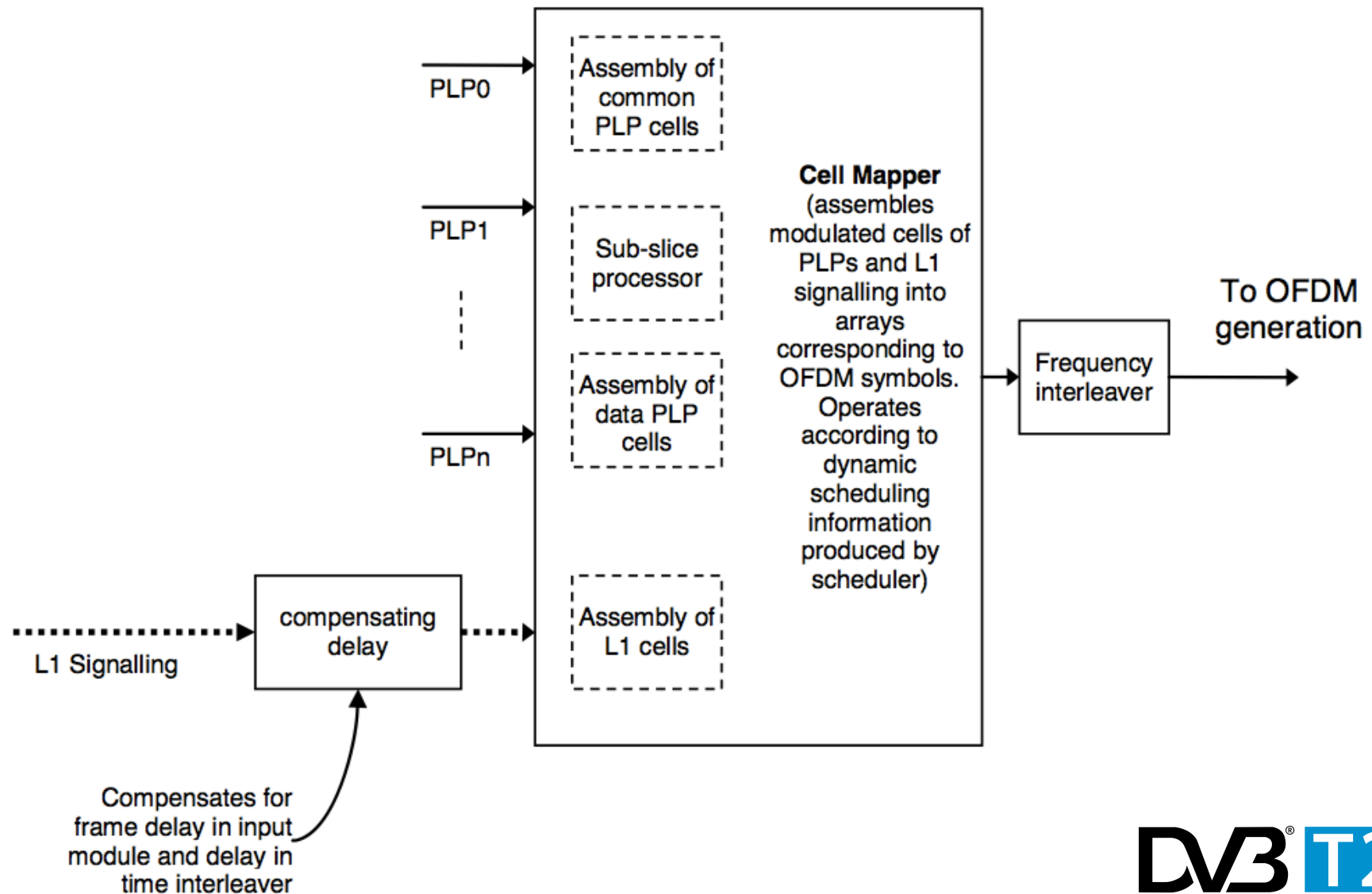


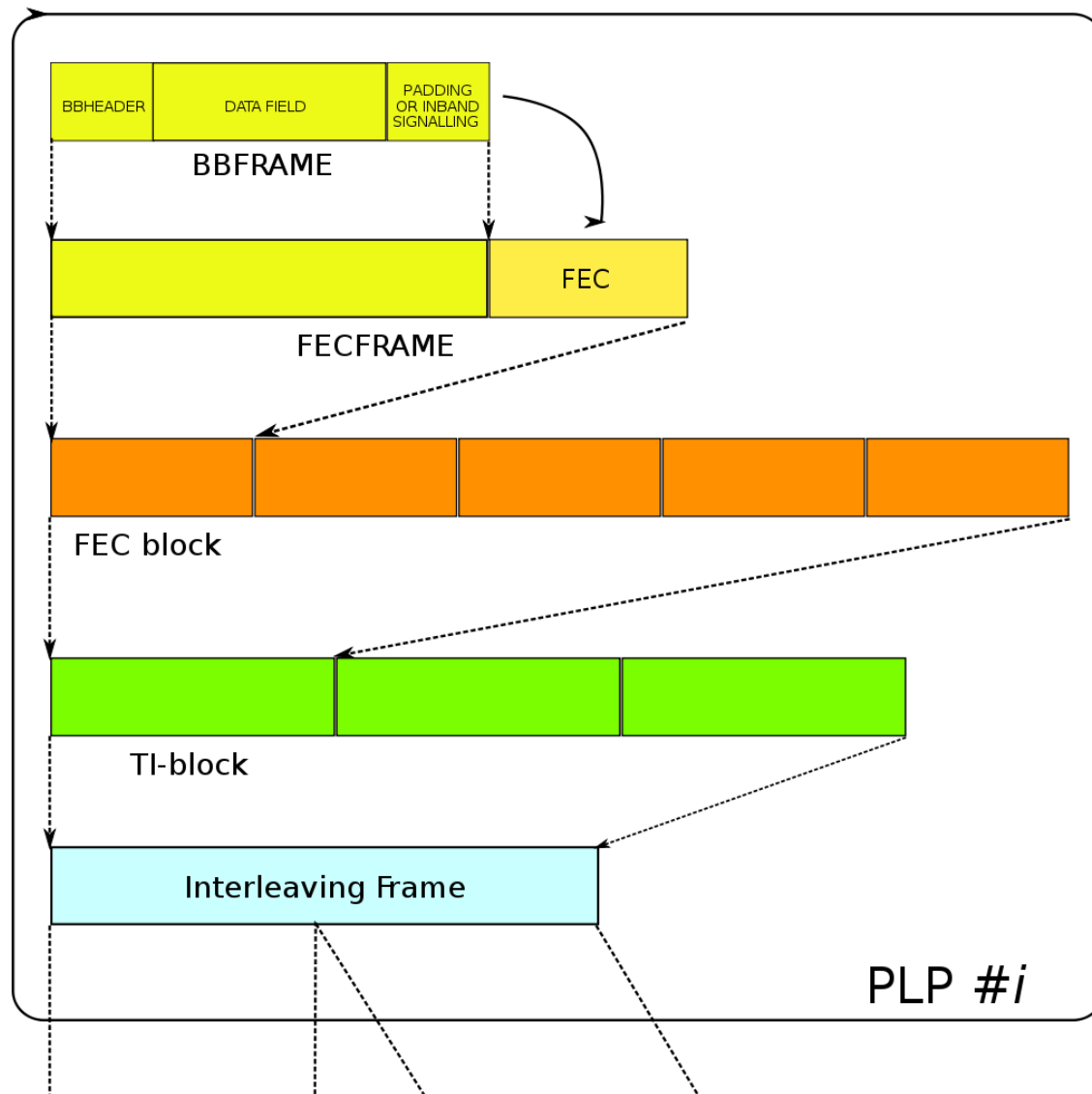


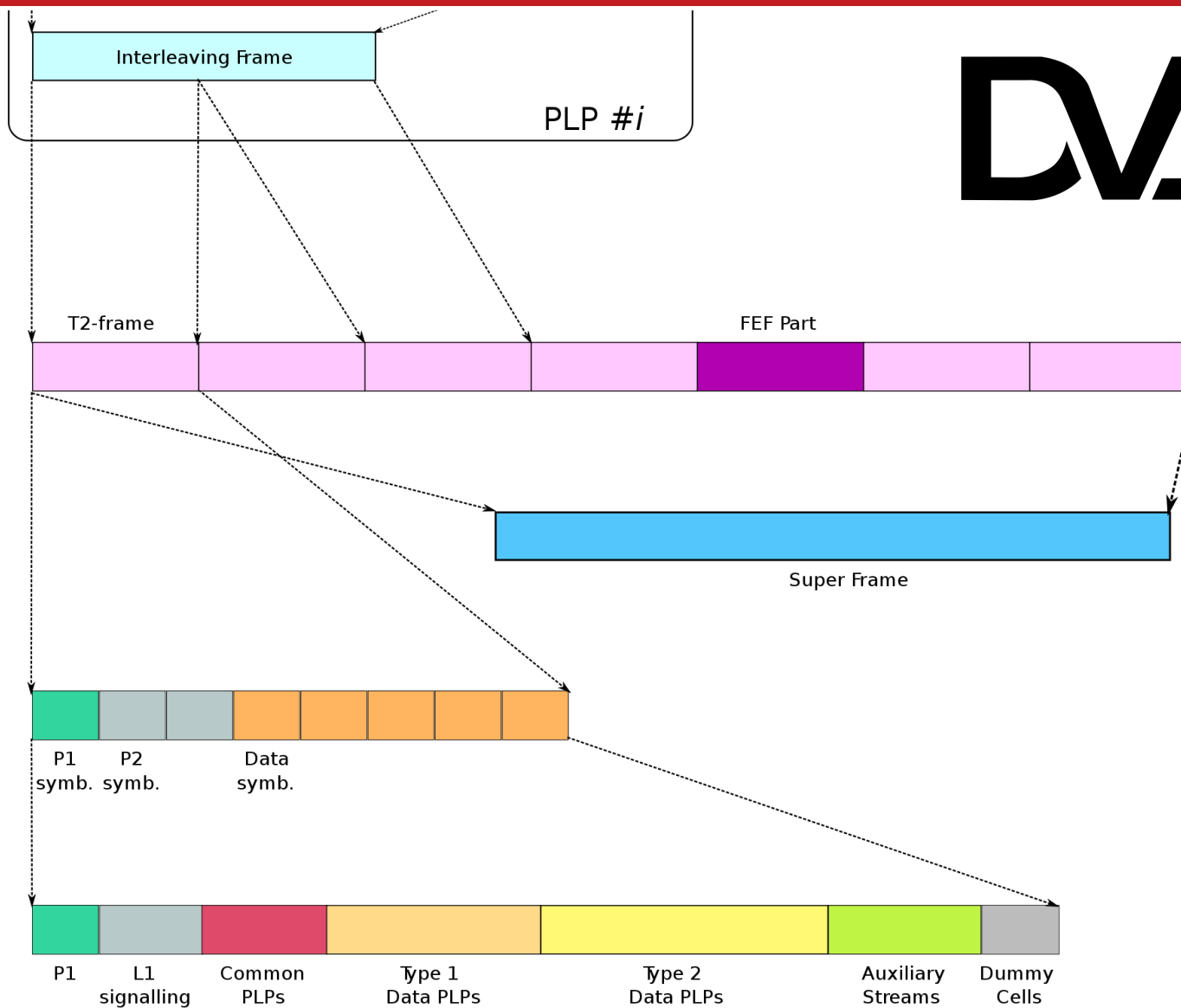
Bit interleaving = parity interleaving + column twist interleaving

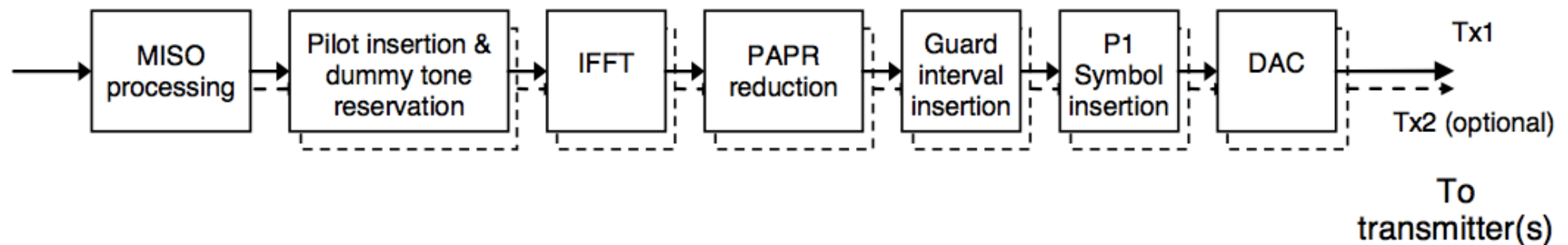


DVB-T2: Frame Builder









- The pilots can be used for frame synchronization, frequency synchronization, time synchronization, channel estimation, transmission mode identification and can also be used to follow the phase noise.
- PAPR = Peak to Average Power Ratio.



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DVB-T2 Transmitter: Full Workflow

- Input pre-processing
 - *Physical Layer Pipe (PLP) creation*: adaptation of Transport Stream (TS), Generic Stream Encapsulation (GSE), Generic Continuous Stream (GCS), or Generic Fixed-length Packetized Stream (GFPS)
- Input processing
 - Mode adaptation
 - *Single PLP (mode 'A')*: data are assembled in groups called BaseBand Frames (BBFRAMEs), with lengths of K_{bch} bits, defined by modulation and coding (MODCOD) parameters, in a 'normal' length or 'short' length version
 - Input interface
 - CRC-8 encoding
 - BaseBand (BB) header insertion
 - Multiple PLPs (mode 'B')
 - Input interface
 - Input stream synchronization
 - Delay compensation
 - Null packets deletion
 - CRC-8 encoding
 - BB header insertion
 - Stream adaptation
 - Single PLP (mode 'A')
 - Padding insertion
 - *BB scrambling*: a Pseudo Random Binary Sequence (PRBS) with generator $1 + x^{14} + x^{15}$ is used to scramble completely every BBFRAME
 - Multiple PLPs (mode 'B')
 - PLP scheduling
 - Frame delay
 - In-band signaling or padding insertion
 - BB scrambling
- Bit Interleaved Coding and Modulation (BICM)
 - *Forward Error Correction (FEC) encoding*: each BBFRAME is converted into a FECFRAME of N_{ldpc} bits, by adding parity data. Normal FECFRAMEs are 64,800 bits long, whereas short FECFRAMEs are 16,200 bits long. The effective code rates are 32,208/64,800 (1/2), 38,688/64,800 (3/5), 43,040/64,800 (2/3), 48,408/64,800 (3/4), 51,648/64,800 (4/5), 53,840/64,800 (5/6)
 - *Outer encoding*: a BCH code, capable to correct 10 or 12 errors per FECFRAME, is used to compute parity data for the information data field. The BCH generator polynomial is of the 160th, 168th, or 192nd grade
 - *Inner encoding*: a Low Density Parity Check (LDPC) code is cascaded to the BCH
 - Bit interleaving
 - Parity bits block interleaving
 - Twist column interleaving
 - Bit demultiplexing to cell words
 - *Gray mapping of cell words to constellations*: either QPSK (4-QAM), 16-QAM, 64-QAM, or 256-QAM maps are used
 - *Constellation rotation and cyclic quadrature (Q) delay*: optionally, the constellations may be tilted counterclockwise by an amount of up to 30 degrees. Furthermore, the quadrature (imaginary) part of the cells is cyclically shifted by one cell
 - Cell interleaving
 - Time interleaving
- Frame building
 - Cell mapping
 - Frequency interleaving
- OFDM generation
 - Multiple-Input Single-Output (MISO) processing
 - Pilot insertion and dummy tone reservation
 - Inverse Fast Fourier Transform (IFFT)
 - Peak-to-Average-Power-Ratio (PAPR) reduction
 - Guard interval insertion
 - P1 symbol insertion
 - Digital-to-Analog Conversion (DAC)



CRC generation

PRBS scrambling

BCH encoding

LDPC encoding

Bit interleaving

Gray mapping

Constellation rot.

Alamouti coding

Inverse FFT

From <http://en.wikipedia.org/wiki/DVB-T2>



- **LDPC encoder**
- **IFFT processor**

All the rest can be implemented with simple(r) hardware.

The design of these core blocks is more critical than in conventional DVB-T since many more modes are defined in terms of code rate and IFFT size.

To support this flexibility in term of modes, we want an implementation that is reconfigurable on the fly (i.e., without downloading new code to the transmitter).



CRC generation

PRBS scrambling

BCH encoding

LDPC encoding

Bit interleaving

Gray mapping

Constellation rot.

Alamouti coding

Inverse FFT

SDR Platform

i.e., the Lyrtech SFF SDR Board

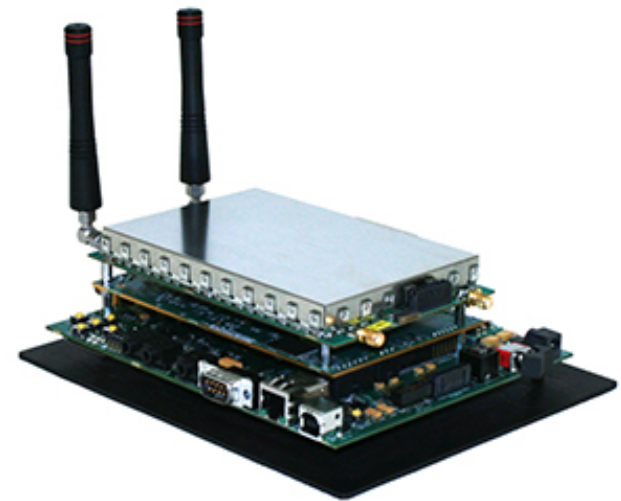


Designed for

- 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

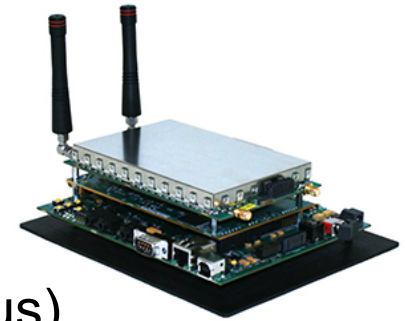
Three distinct modules:

- **RF Module**
- **Data Conversion Module**
- **Digital Processing Module**

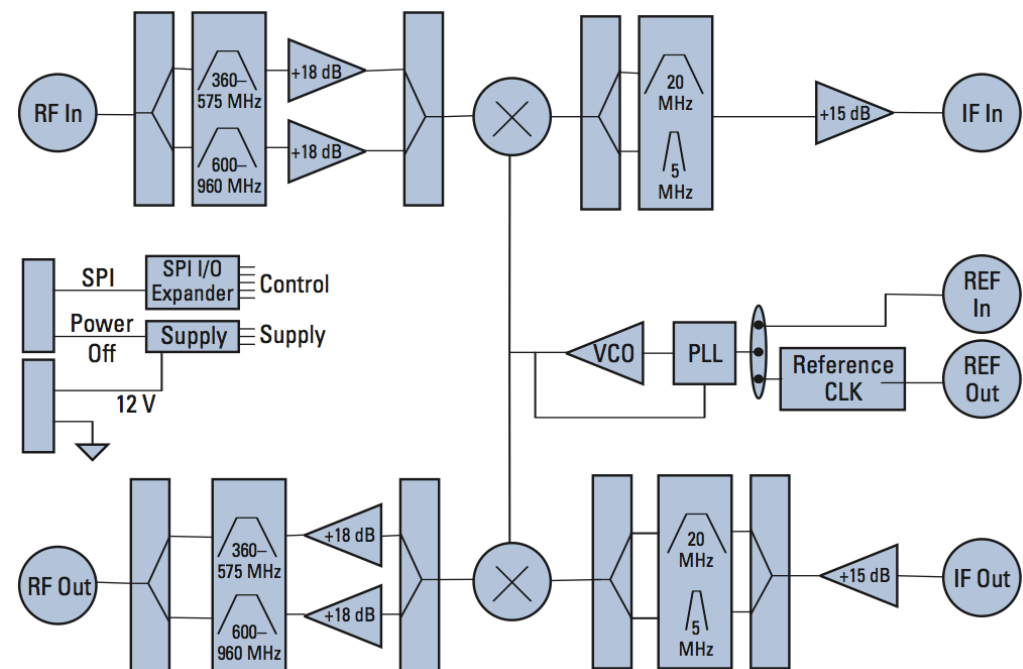




- RF frequency range of 360 MHz to 960 MHz
- Selectable bandwidth: 5 MHz / 20 MHz
- Separate RF input and output
- Phase noise at 20 kHz from carrier: -70 dBc
- Half-duplex transceiver; stackable (on proprietary LYRIO bus) for full-duplex operation



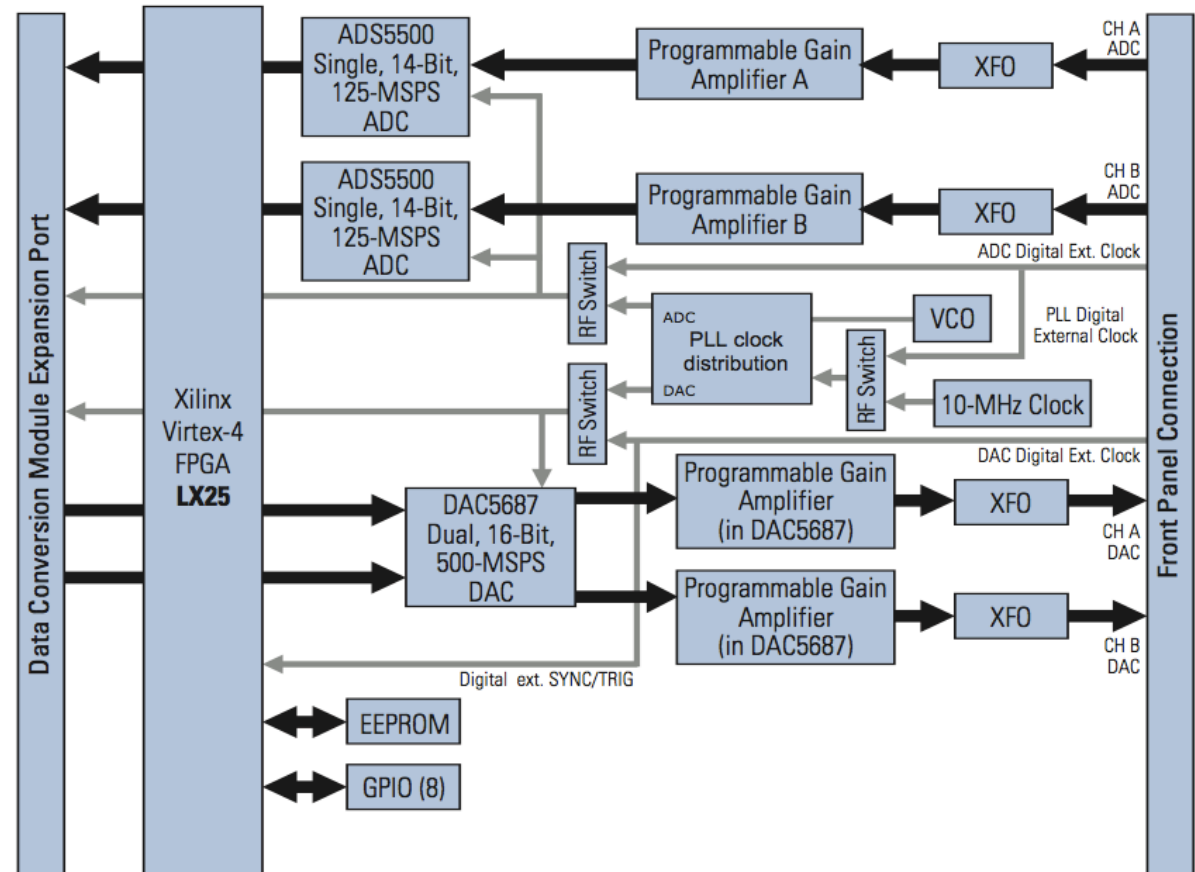
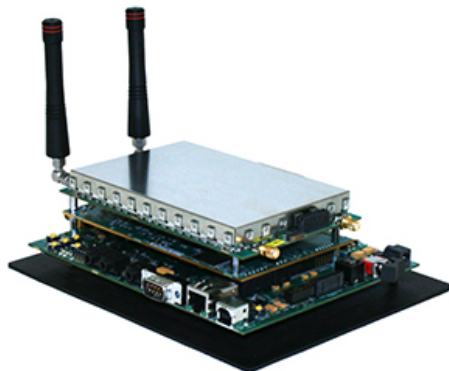
- RF input
 - Gain: 22 dB
 - Saturation level: -30 dBm
 - Sensitivity: -110 dBm typical
- RF Output
 - Gain: 22 dB
 - Power: -5 dBm





Data Conversion Module

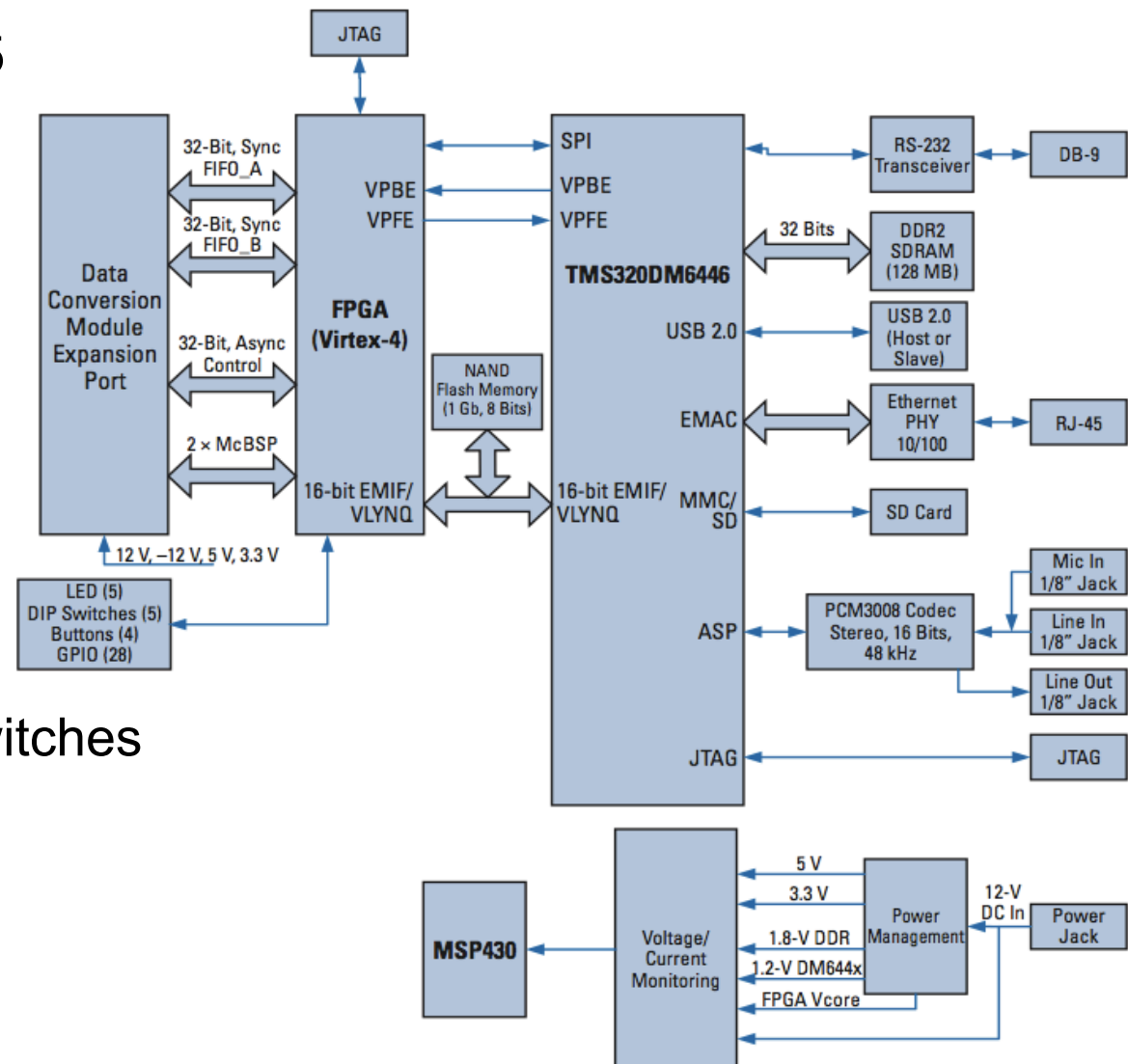
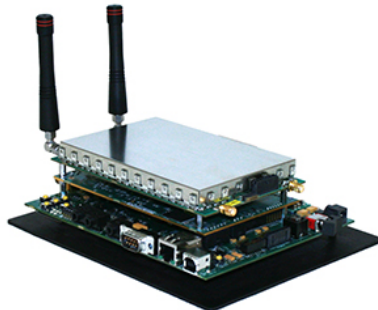
- Two 14-bit, 125-MSPS input channels
- Dual-channel 16-bit, 500-MSPS output channels
- Multiple clock sources
 - Two ext. clock inputs (ADC and DAC)
 - 10-MHz onboard reference clock
 - Reference clock input for synchronization






Digital Processing Module

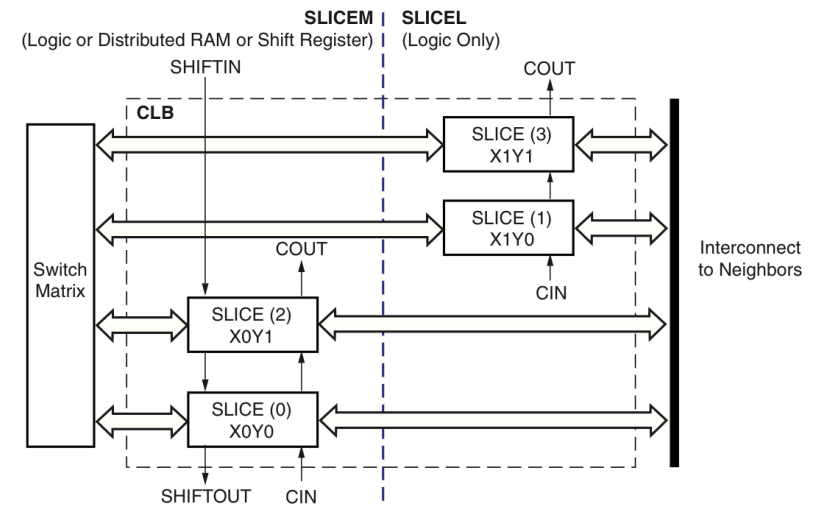
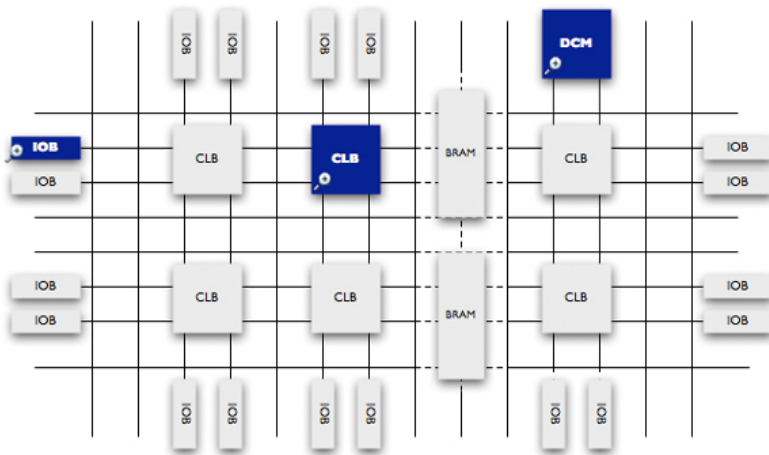
- FPGA: Xilinx Virtex-4 SX35
- GPP+DSP: TI “DaVinci”
- MCU: TI MSP430 MCU
- 128-MB DDR2 SDRAM
- 128-MB NAND Flash
- Stereo audio codec
- 10/100-Mbit/s Ethernet
- JTAG probing access
- HMI: LEDs, buttons, dip switches





FPGA: Xilinx Virtex4 SX35

- Array of Configurable Logic Blocks (CLBs) : 96 rows x 40 columns
 - Number of slices: 15,360
 - Number of Look-Up Tables (LUTs): 30,720
 - Maximum distributed RAM or shift registers: 240 kilobits
 - Number of flip-flops: 30,720
- 



(Much) more about this in tomorrow's talk.



Texas Instruments “DaVinci” SoC

ARM926EJ-S core @297 MHz

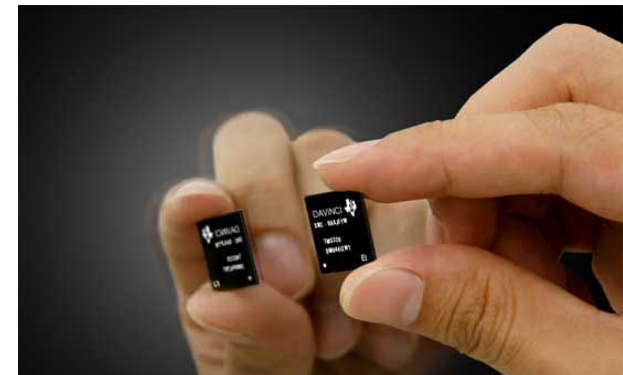
- Co-Processor 15
- MMU
- 16 KB I-cache, 8KB D-cache
- 16KB Internal RAM
- 8KB Internal ROM

TMS320C64x+ DSP core @ 594 MHz

- 2 datapaths
- 8 functional units, each capable of executing 1 instruction per cycle
- 32KB L1 Program (L1P)/Cache
- 80KB L1 Data (L1D)/Cache
- 64KB Unified Mapped RAM/Cache

Rich set of peripherals: UARTs, USB, Ethernet, memory card interface, SPI, VLIQ™, Video Processing Subsystem...

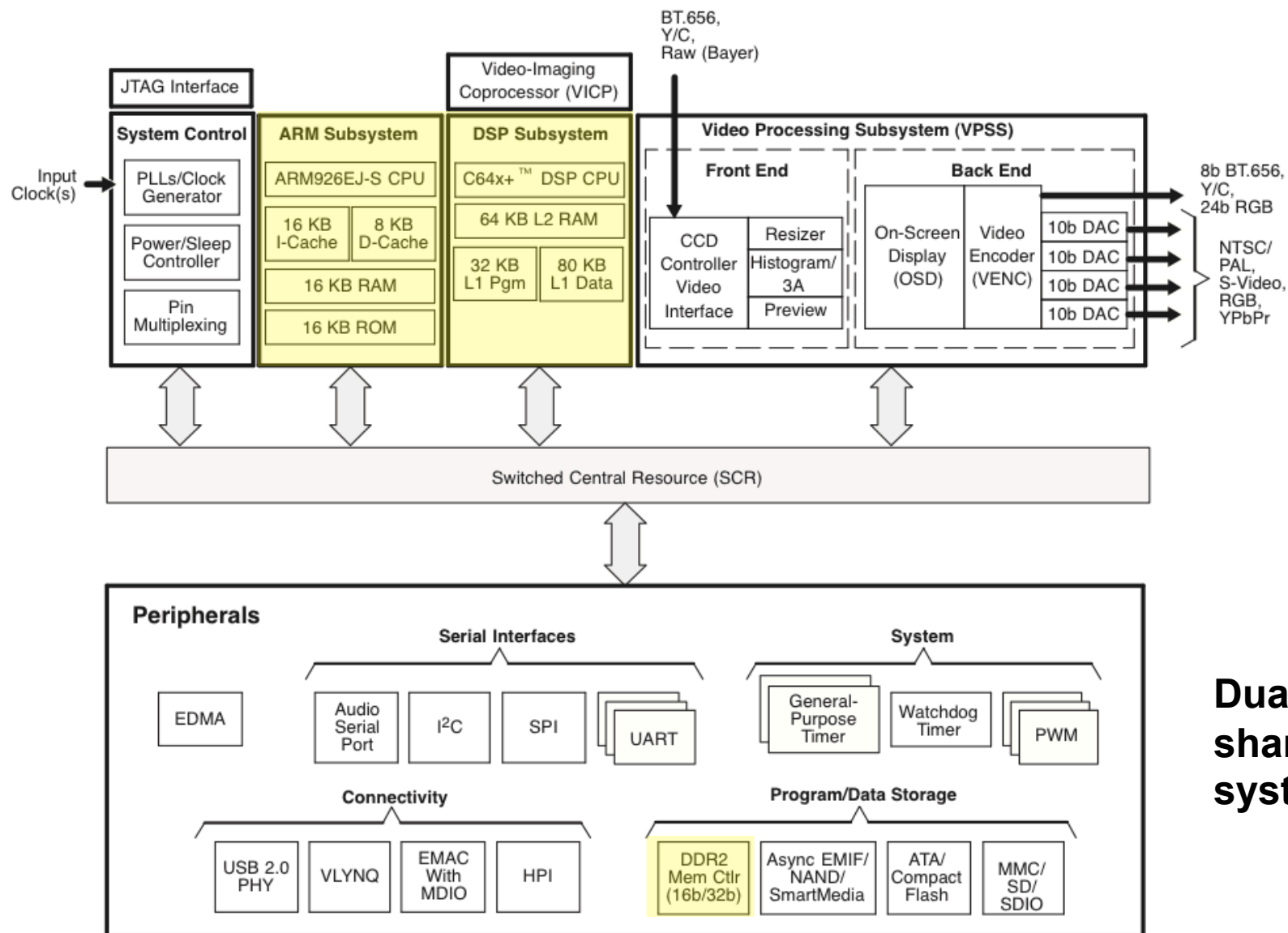
This is why TI calls it a “system-on-Chip” (SoC).





UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DaVinci: Functional Block Diagram



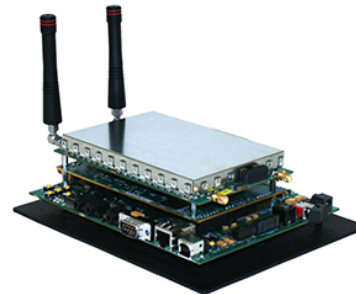
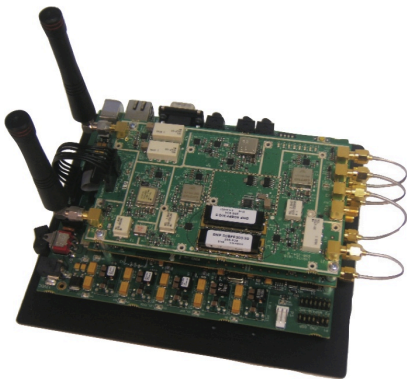
**Dual-core,
shared-memory
system**



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

SFF SDR: Software Development Tools

- Texas Instruments Code Composer Studio IDE and libraries
- Xilinx ISE Foundation for FPGA development
- Xilinx System Generator for DSP
- Green Hills Software MULTI® IDE
- Green Hills Software POSIX-compliant INTEGRITY® real-time OS
- MontaVista Linux
- The MathWorks MATLAB® and Simulink
- Software Communication Architecture (SCA) tools available



Software



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Software: ARM Core

The **ARM9** core is responsible for configuration and control of the DaVinci and, consequently, of the whole **SFF SDR board**, with particular reference to

- the DSP subsystem,
- the peripherals and memories,
- the FPGA.

The SFF SDR board ships programmed with an evaluation version of the **INTEGRITY** RTOS.

It is closed-source software: in particular, it is not possible to modify and recompile the **INTEGRITY** kernel. This was **unacceptable** for our needs.

→ **Switch to U-Boot and Linux**





Das U-Boot: the Universal Boot Loader

- Project initiated and hosted by DENX
- Open-Source: GPL license
- Supports many boards based on Power®, ARM, MIPS, x86 and other processors
- Allows boot from Flash memory or network

See <http://www.denx.de/wiki/U-Boot> for more info.

SFF SDR board supported out of the box.



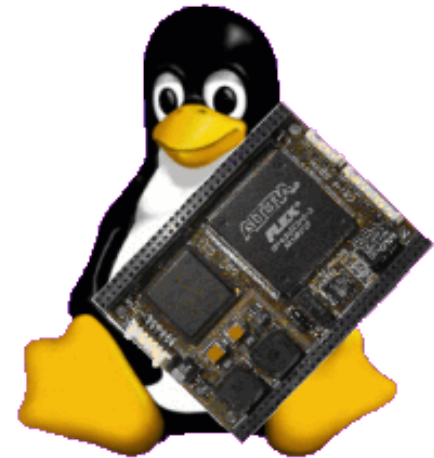


MontaVista Linux 4.0

- DaVinci: the VPSS (Video Processing SubSystem, (VPSS, more on this later) is fully supported
- Based on an outdated kernel version (2.6.10)
- Proprietary patches, will never be integrated in the main kernel tree

Mainline kernel (2.6.32)

- Texas Instruments (→Kevin Hilman) is pushing patches for the DaVinci into the main kernel tree
- Flash memory support has been rewritten
- USB works (in MontaVista it does not)
- Support for the VPSS is incomplete, but things are improving over time and the drivers fit the new Video4Linux device model



We adopted the Linux kernel tree maintained by Kevin Hilman:
<http://git.kernel.org/?p=linux/kernel/git/khilman/linux-davinci.git>



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

The OpenEmbedded Project

The kernel is not enough: a full distribution is needed, where standard components (shell, scripting languages, libc, ...) are available, and where additional libraries can be added.

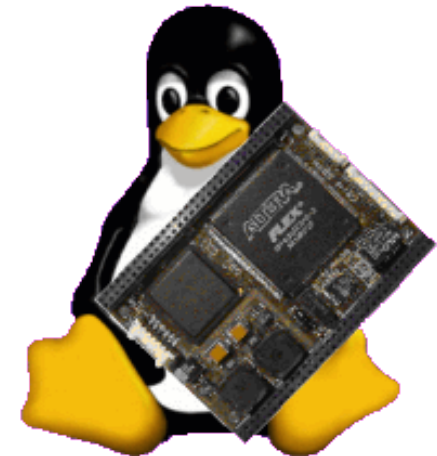
OpenEmbedded: a software framework to create Linux distributions aimed for embedded devices.

Hundreds of components are available via “recipes”, i.e. scripts that define

- where to download the sources of each component,
- which patches to apply,
- which steps must be performed to compile the sources.

Build a distribution = select needed recipes.

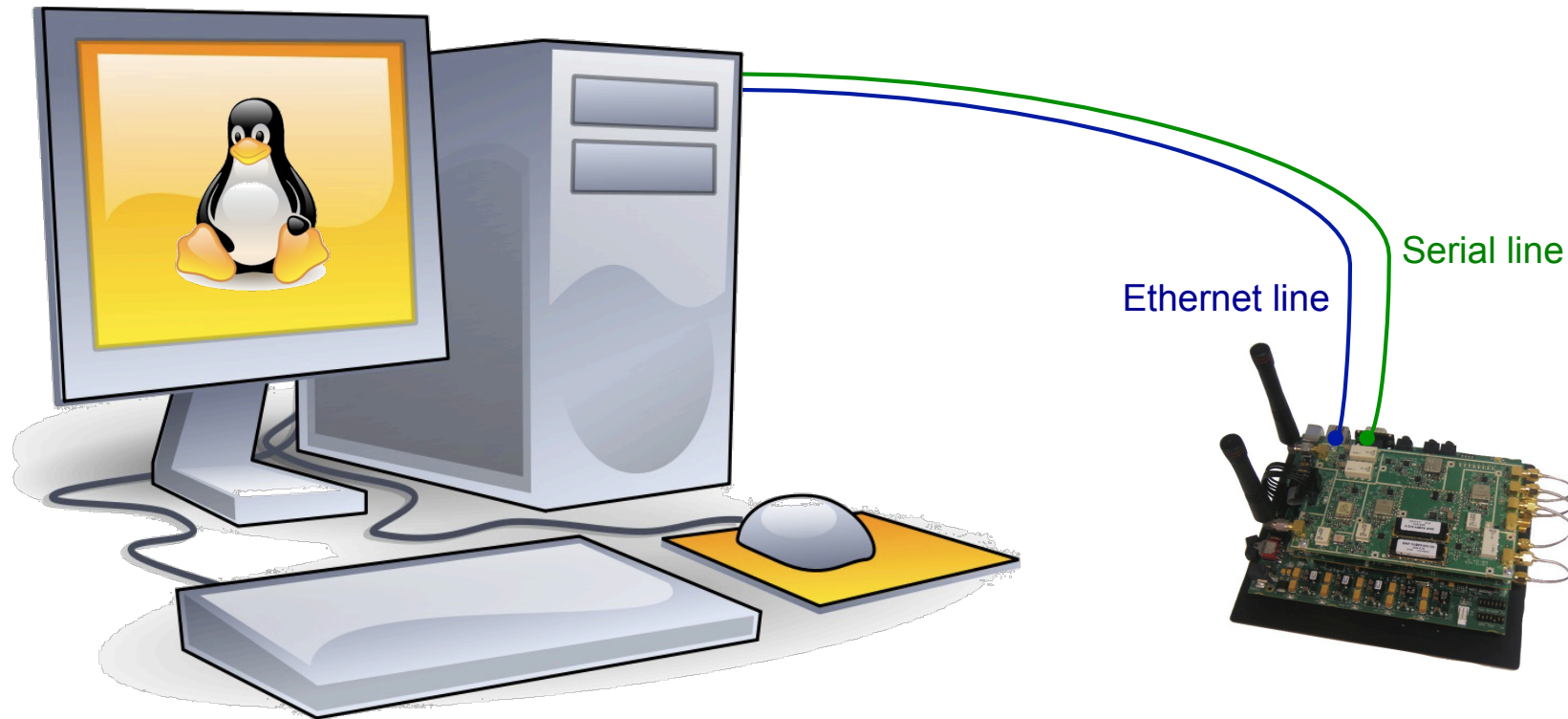
Important: **recipes exist for standard TI libraries.**





UNIVERSITÀ
DEGLI STUDI
DI PADOVA

SFF SDR: Development Station

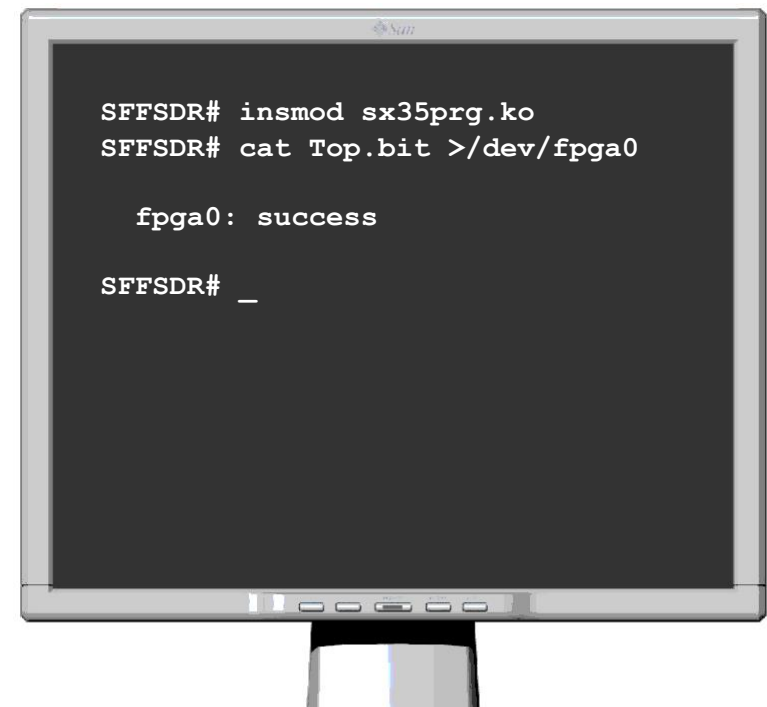
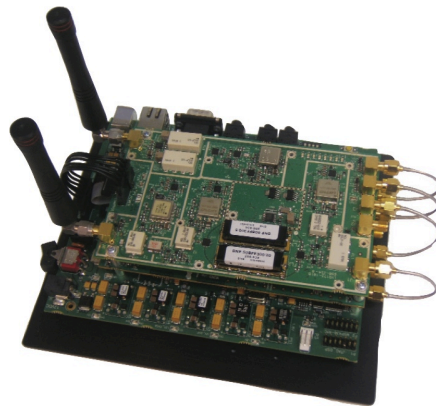


- Some Linux distro
- OpenEmbedded
- Cross-compilers for ARM and C64x+
- Libraries

- Kernel loaded via **TFTP**
- Filesystem mounted via **NFS**
- Console: via **serial line**



A small (≈ 400 LOC) Linux kernel driver has been written to send the bitstream (.bit) to the FPGA without the need for an external JTAG programmer.



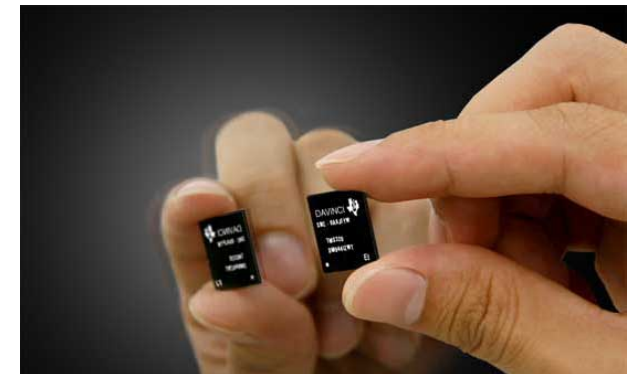


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DaVinci's DSP: Code Generation Tools

The TMS320C6000 C/C++ Code Generation Tools is a package containing **optimizing compilers for TI DSPs in the C6xxx family.**

- C and C++ languages
- Support for several GCC extensions
- C++ Standard Template Library (STL)
- Profile-based optimization



Proprietary software developed by TI.

Can be freely downloaded for Windows and Linux.
A recipe exists in OpenEmbedded.



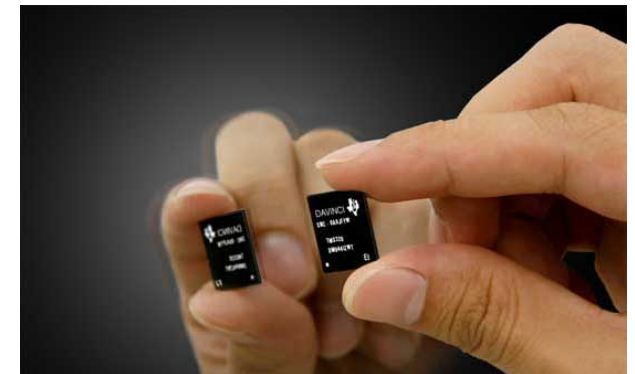


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DSP components: DSP/BIOS

DSP/BIOS is a **real-time OS for the C64x+ DSP core** found inside the DaVinci chip. It provides:

- task creation and pre-emptive multitasking services,
- management of hardware and software interrupts,
- semaphores, pipes, queues, messages,
- device I/O,
- memory management,
- power management,
- debug instrumentation and statistics gathering.



Proprietary software developed by TI.

- Runtime: no fees. Language to use: C.
A recipe exists in OpenEmbedded
- Source code: significant fees + NDA to access it.
Luckily, we do not need the source

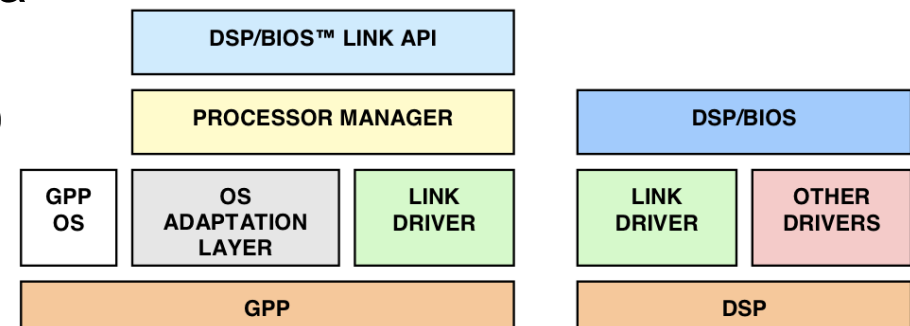




DSP components: DSPLink

DSP/BIOS Link (or simply “DSPLink”) is a software library that provides **GPP-to-DSP communications and control services**.

- Basic processor control
- Shared/synchronized memory pool across multiple processors
- Notification of user events
- Mutually exclusive access to shared data
- Data transfer over logical channels
- Messaging (even zero-copy messaging)
- Linked-list-based data streaming
- Ring-buffer-based data streaming



Proprietary software developed by TI.

A recipe exists for OpenEmbedded.

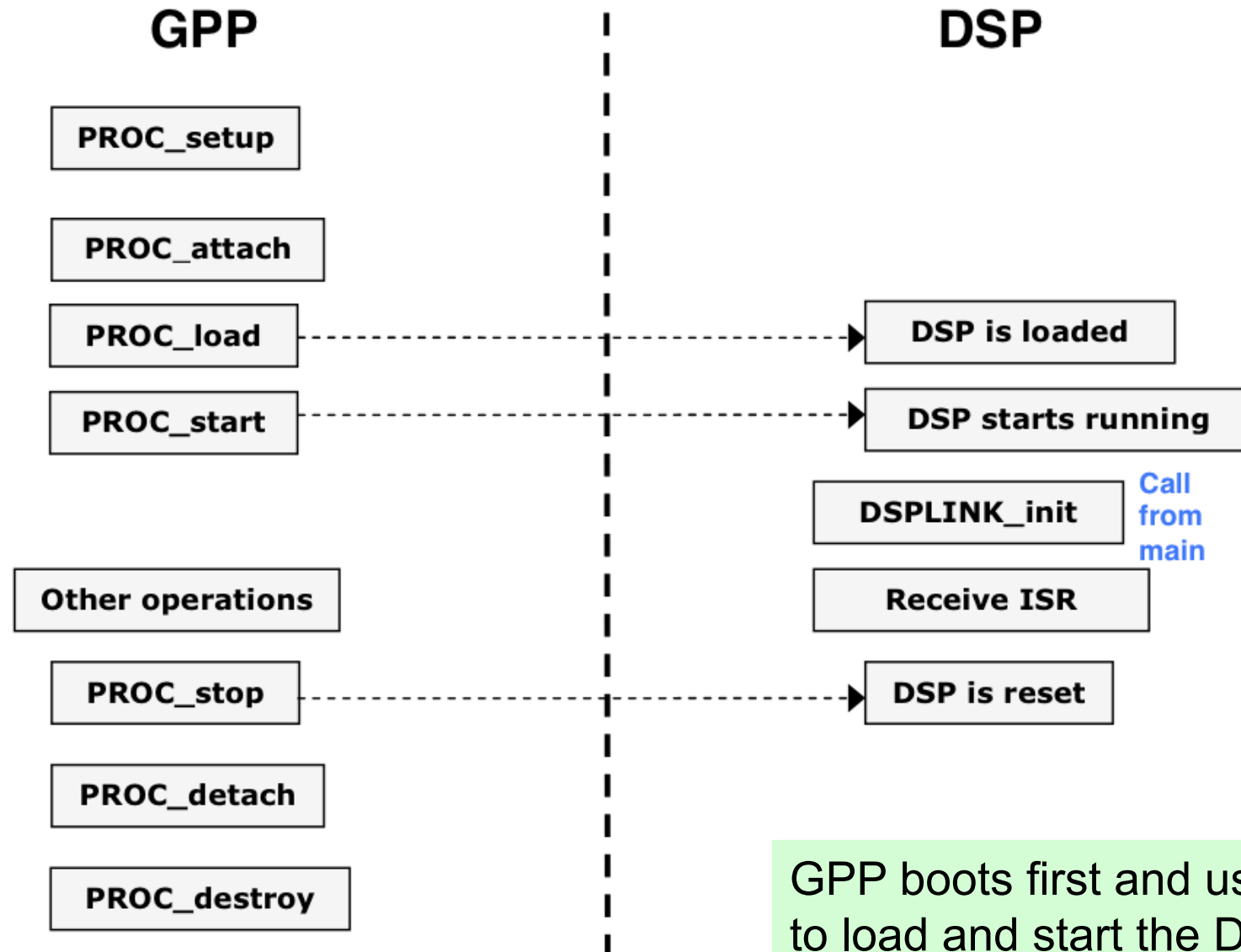
The DSP/BIOS OS is required on the DSP.

No specific OS is mandated to be running on the GPP.





DSPLink: Operation



GPP boots first and uses DSPLink to load and start the DSP core



Fundamental design decision: **where to implement the core components** of the transmitter:

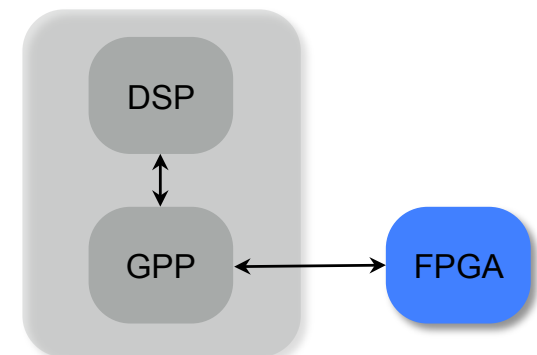
- LDPC encoder
- IFFT processor



Available units: GPP, DSP, FPGA.

First step: calculating the maximum data rate, to be sure that it can be sustained in terms of

- computing resources
- available bandwidth among units





Worst case:

- FFT size: **32,768** points
- Minimal number of pilots → 27,404 active cells
→ **219,232** bits per symbol
- Symbol period: **3.584** ms
- Guard Interval (GI): $1/128 \approx 0$

→ **Data rate: \approx 61 megabits per second**

FFT size	T_u (ms)
32K	3.584
16K	1.792
8K	0.896
4K	0.448
2K	0.224
1K	0.112



- **GPP**: ARM926EJ-S core
- Clocked at 297 MHz



Even assuming (optimistically) that the GPP is able to execute one instruction per cycle, at the maximum data rate less than 5 instructions per bit are available for computation.

→ **Neither the LDPC encoder nor the IFFT can be implemented on the GPP**





- **DSP:** T64x+ core
- Clocked at 594 MHz
- Peak performance:
 - 4,752 Million Instructions Per Second (MIPS)
 - 96 Billion Multiply-and-Accumulates per Second (GMACS)



➔ Both LDPC and IFFT can theoretically be run on the DSP

- **LDPC:** literature shows **no implementations** because of the structure of LDPC computation
- **IFFT:** a library from TI (DSPLIB) exists



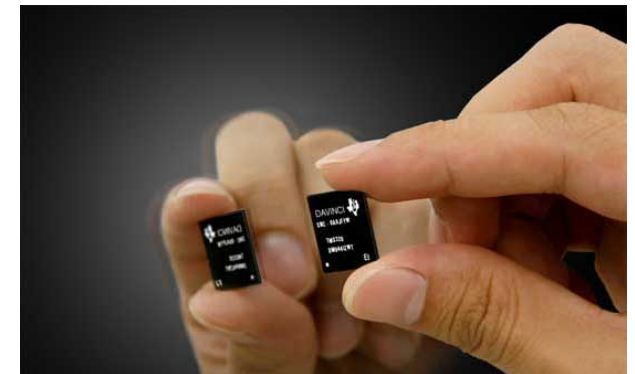


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DSP Components: DSPLIB

The TMS320C64x+ DSP Library (or simply “DSPLIB”) is a an **assembly-optimized software library** for C programmers using devices with the C64x+ core. It contains **fast routines for common DSP tasks**:

- FIR filtering and convolution,
- adaptive FIR filtering,
- correlation,
- FFT and IFFT,
- math operations on vectors,
- math operations on matrices.



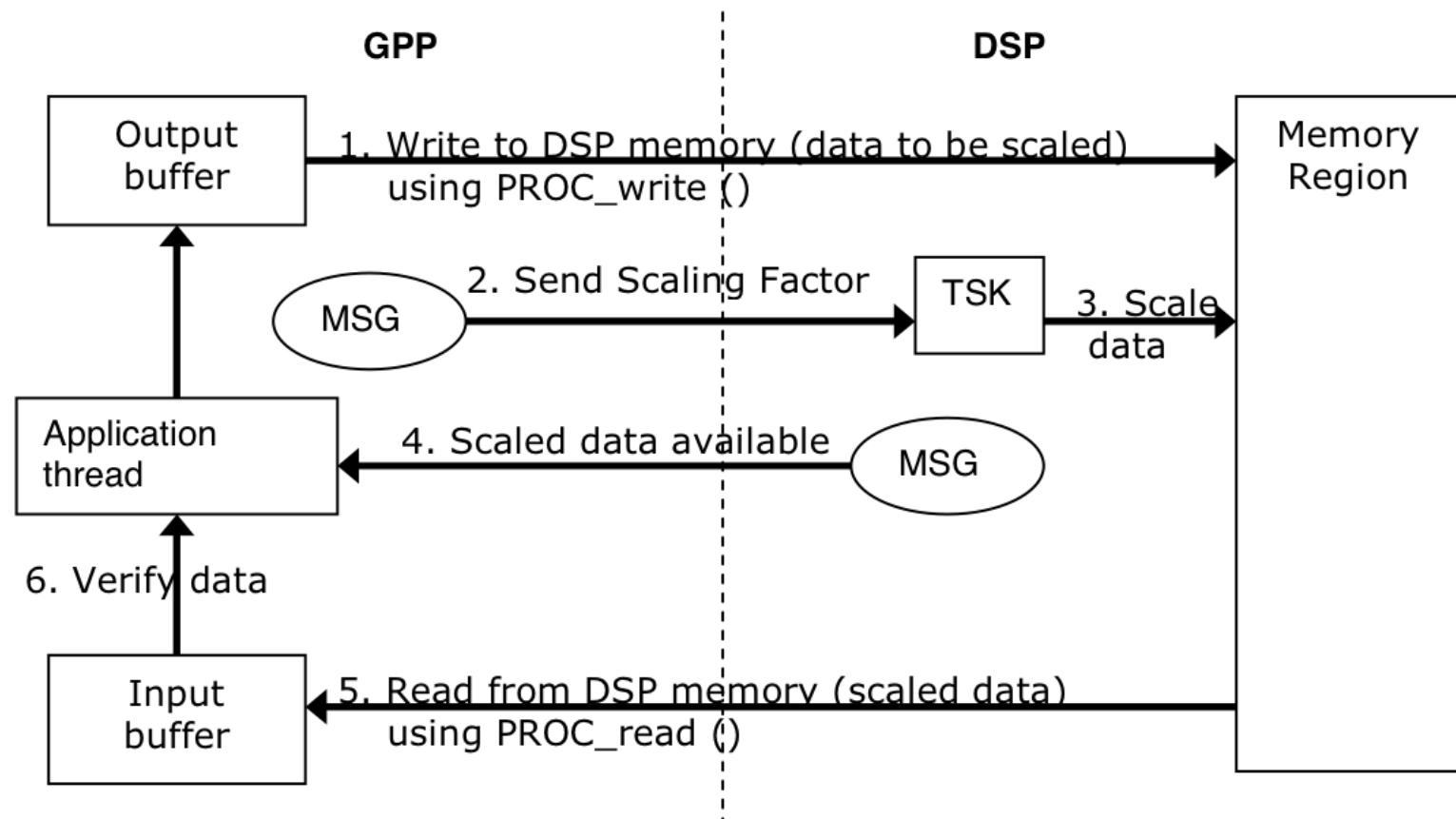
Proprietary software developed by TI.
No recipe for OpenEmbedded.





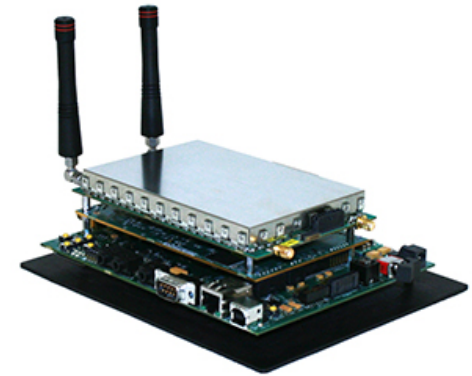
Communication between GPP (ARM) and DSP

- Only one reasonable solution: **shared memory**
- Perfectly supported by DSPLink



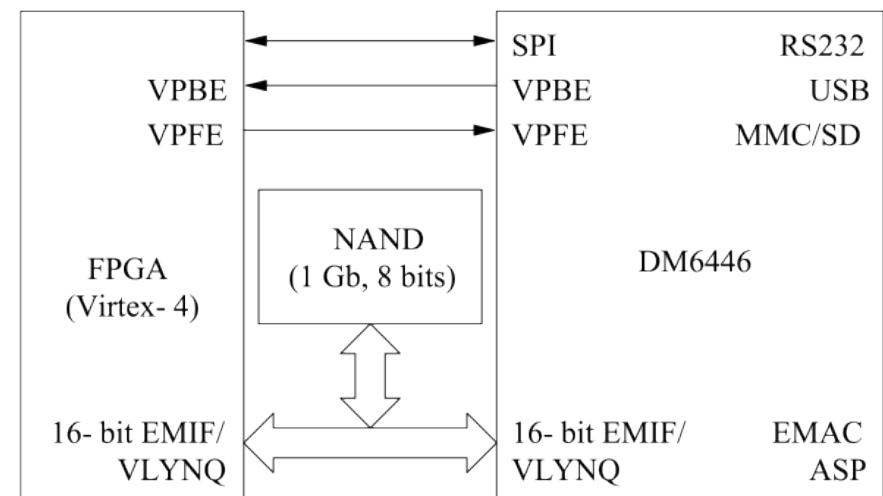


Communication between DaVinci and FPGA:
different solutions, with different characteristics.



We will consider 4 of them:

- SPI bus,
- VLYNQ™ interface,
- EMIF interface,
- **VPSS** (innovative solution).



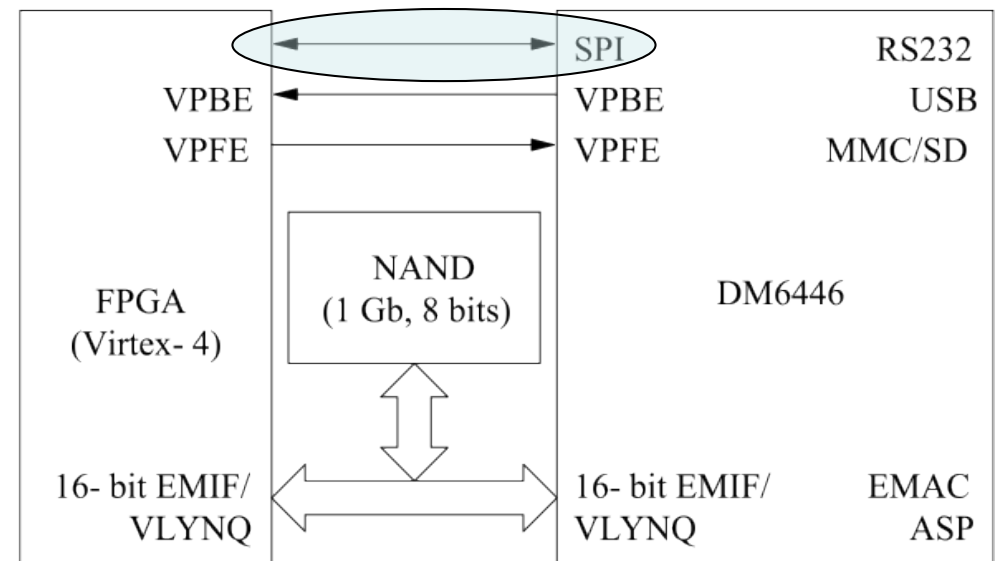


SPI = **S**erial **P**eripheral **I**nterface

- Industry-standard, simple (4-wire) interconnection
- Full duplex

On the DaVinci chip:

- Can be operated via DMA
- Maximum clock: **8.3 MHz**



➔ Maximum throughput: 16.6 Mbit/s

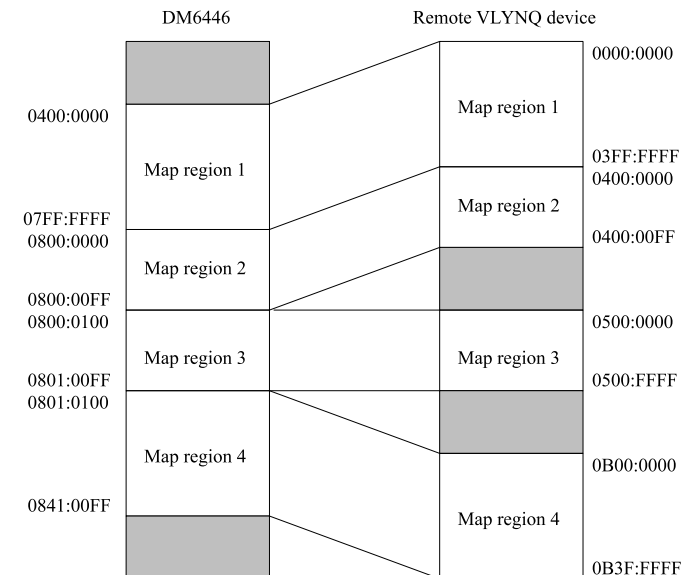
➔ **Cannot be used** for our purposes

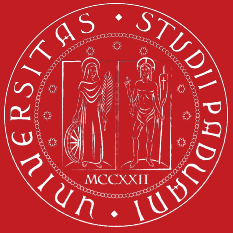


- High-speed, low-pin-count interface
- Proprietary
 - Developed by TI
 - Xilinx IP available for the FPGA

On the DaVinci chip

- Mutually exclusive with EMIF
- Seen by DaVinci as a memory area shared with the FPGA
- Max throughput: 584 Mbit/s
- **No support** in the vanilla Linux kernel
- Support in MontaVista Linux proved to be **unreliable**
- Support from Xilinx **discontinued** as of 10/31/2009





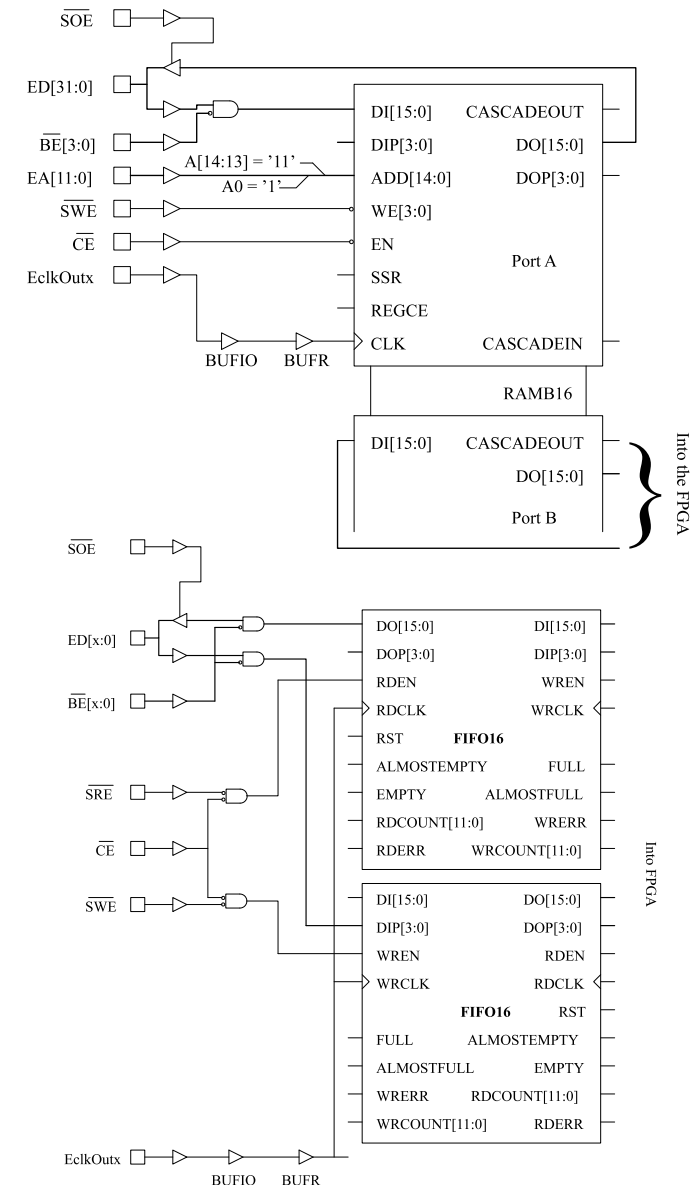
EMIF = External Memory InterFace:
Provided by TI to connect a variety of
Flash and random access memories.

On the DaVinci chip

- Mutually exclusive with VLIQTM
- Address bus: 24 bit (16 MB addressable)
- Data bus: 8 bit or 16 bit

Interfacing with Xilinx FPGAs

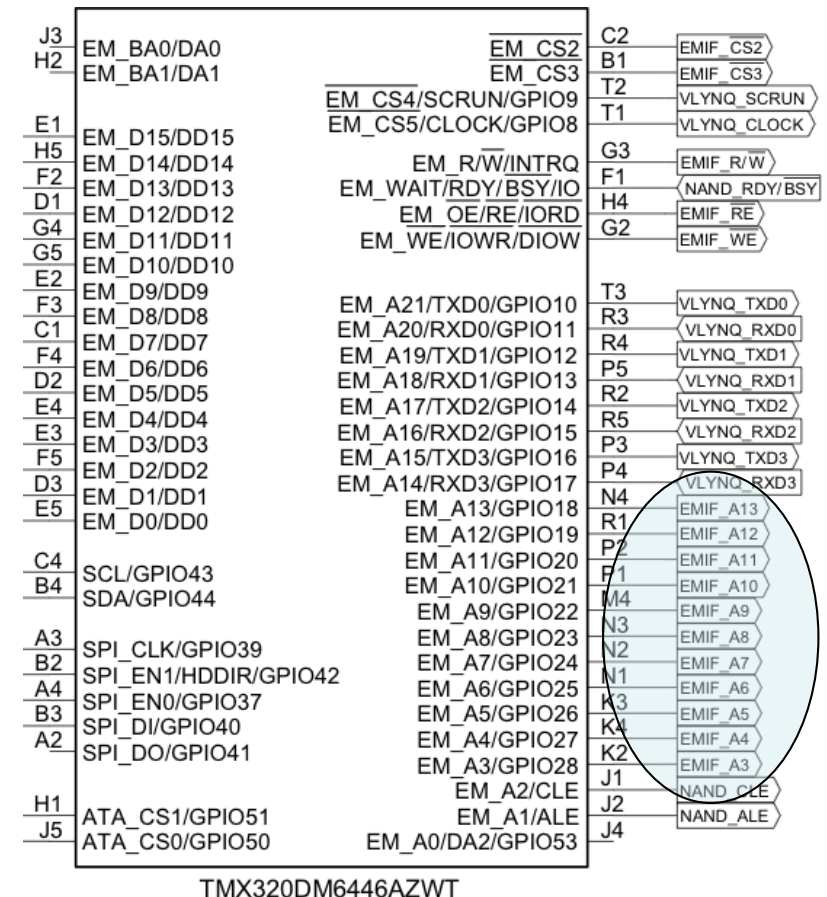
- Described in Xilinx XAPP 753
(but DaVinci requires customizations)
- First approach: uses dual port block RAMs
- Second approach: uses FIFOs





Our tests showed that on the SFF SDR board EMIF exhibits **limitations that advise against its usage for our purposes**

1. Communication from FPGA to DaVinci: some unconnected pins make communication cumbersome and slow
2. Address lines are connected in a way that make consecutive bytes appear many bytes apart in the EMIF region: a significant number of GPP cycles must be spent to remap the bytes, thus lowering the throughput to an unacceptable value



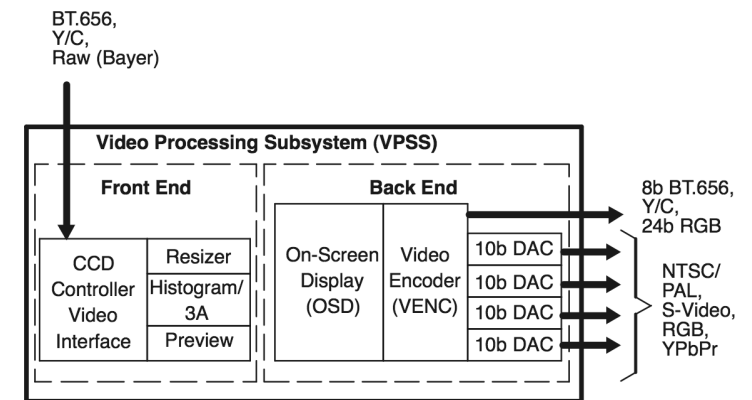
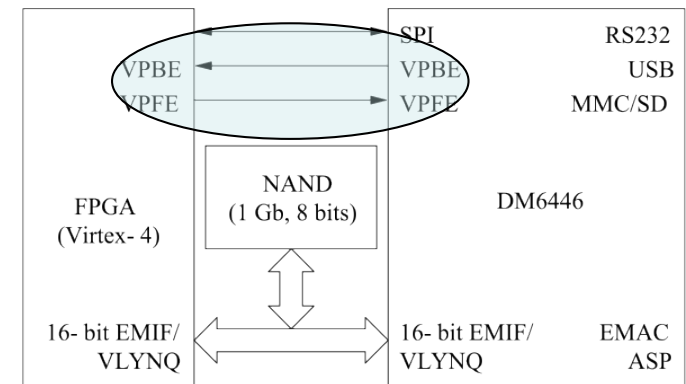


VPSS = Video Processing SubSystem

- TI proprietary
- VPFE = Video Processing Front End: video input (from the external world into DaVinci)
- VPBE = Video Processing Back End: video output
- Digital video I/O using standard BT.656 format
- Bit rate: **≈83 Mbit/s** (PAL format, 8 bits per pixel)
- Very low overhead for the ARM core
- Video for Linux 2 (V4L2) support

Idea: **data can be transmitted instead of pixels** by simulating a fake video device (with suitable coding in the FPGA and in the Linux kernel).

→ **We adopted the VPSS** for communication between the FPGA and the DaVinci chip





Our experience with DSPLIB

- For $n = 8,192$ points, $T_{\text{IFFT}} = 0,55$ ms
- For $n > 8,192$ points, T_{IFFT} explodes (cache effects)
- Times do not include overheads
(GPP \leftrightarrow DSP sync, bit manipulations, ...)



Recall that, because of the symbol rate,
we must produce

- a 32,768-point IFFT every 3.584 ms,
- an 8,192-point IFFT every 0.896 ms.



→ Tight margins, **better to look for another solution.** (?)

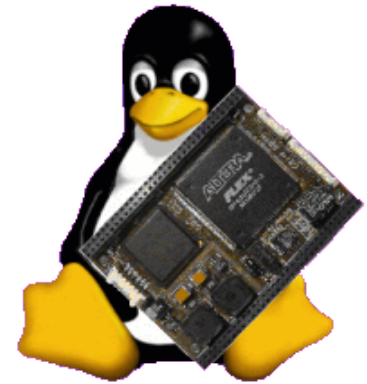


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Software Summary

- Linux kernel
- OpenEmbedded + BitBake
- DSP/BIOS OS
- DSPLink library
- DSPLIB library

...Plus software written for the project





Conclusions: despite the fact that the DVB-T2 standard poses higher challenges in terms of computational complexity, the transmitter can be implemented on commercially-available SDR platforms.

Future work: completing the transmitter with the stream manipulation, signaling and scheduling functions mandated by the DVB-T2 standard. Since two powerful cores (an ARM926EJ-S GPP running at 297 MHz and a TMS320C64x+ DSP clocked at 567 MHz) are still almost free and therefore available for the aforementioned functions, we are confident that the full system can be implemented on the chosen SDR board.





UNIVERSITÀ
DEGLI STUDI
DI PADOVA

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

Carlo Fantozzi, Assistant Professor, DEI

Brixen, June 30 2010