Digital circuits for test and slow control of ASICs in radiation environment

Università degli Studi di Torino,
Laurea Magistrale in in Fisica delle Tecnologie Avanzate

Serena Panati

Wednesday, April 17th, 2013

Introduction to design for testability

Design for testability (DFT) is a process that incorporates **rules** and **techniques** in the design of a device to make testing easier.

It is a system methodology that impacts all phases of a device's life and is used to **manage complexity**, to **minimize development time** and to **reduce manufacturing costs**.

Testing has two major aspects: **control** and **observation**.

To test any system it is necessary:

Introduction to design for testability

Design for testability (DFT) is a process that incorporates **rules** and **techniques** in the design of a device to make testing easier.

It is a system methodology that impacts all phases of a device's life and is used to **manage complexity**, to **minimize development time** and to **reduce manufacturing costs**.

Testing has two major aspects: **control** and **observation**.

To test any system it is necessary:

- to put the system into a known state

Introduction to design for testability

Design for testability (DFT) is a process that incorporates **rules** and **techniques** in the design of a device to make testing easier.

It is a system methodology that impacts all phases of a device's life and is used to **manage complexity**, to **minimize development time** and to **reduce manufacturing costs**.

Testing has two major aspects: **control** and **observation**.

To test any system it is necessary:

- to put the system into a known state
- to supply known input data (*test data*) and state

Introduction to design for testability

Design for testability (DFT) is a process that incorporates **rules** and **techniques** in the design of a device to make testing easier.

It is a system methodology that impacts all phases of a device's life and is used to **manage complexity**, to **minimize development time** and to **reduce manufacturing costs**.

Testing has two major aspects: **control** and **observation**.

To test any system it is necessary:

- to put the system into a known state
- to supply known input data (*test data*) and state
- to observe the system to see if it performs as designed and manufactured

Benefits Over Standard Test Methods

Increased complexity and lack of physical access to modern circuitry makes for costly and time-consuming testing using traditional test techniques.

Adding **testability** to a device:

A structured technique such as **Boundary-Scan Testing** allows for pins-out testing to easily detect these failures.
The idea behind of **incorporate design-for-test techniques** is to:

Benefits Over Standard Test Methods

Increased complexity and lack of physical access to modern circuitry makes for costly and time-consuming testing using traditional test techniques.

Adding **testability** to a device:

- **increases** design time and costs

A structured technique such as **Boundary-Scan Testing** allows for pins-out testing to easily detect these failures.
The idea behind of **incorporate design-for-test techniques** is to:

Benefits Over Standard Test Methods

Increased complexity and lack of physical access to modern circuitry makes for costly and time-consuming testing using traditional test techniques.

Adding **testability** to a device:

- **increases** design time and costs
- **reduces** costs of design validation, manufacturing test, and system maintenance.

A structured technique such as **Boundary-Scan Testing** allows for pins-out testing to easily detect these failures.
The idea behind of **incorporate design-for-test techniques** is to:

Benefits Over Standard Test Methods

Increased complexity and lack of physical access to modern circuitry makes for costly and time-consuming testing using traditional test techniques.

Adding **testability** to a device:

- **increases** design time and costs
- **reduces** costs of design validation, manufacturing test, and system maintenance.

A structured technique such as **Boundary-Scan Testing** allows for pins-out testing to easily detect these failures.
The idea behind of **incorporate design-for-test techniques** is to:

- allows test instructions and data to be serially loaded into a device and enables the subsequent test results to be serially read out.

Benefits Over Standard Test Methods

Increased complexity and lack of physical access to modern circuitry makes for costly and time-consuming testing using traditional test techniques.

Adding **testability** to a device:

- **increases** design time and costs
- **reduces** costs of design validation, manufacturing test, and system maintenance.

A structured technique such as **Boundary-Scan Testing** allows for pins-out testing to easily detect these failures.
The idea behind of **incorporate design-for-test techniques** is to:

- allows test instructions and data to be serially loaded into a device and enables the subsequent test results to be serially read out.
- allows simpler access to circuits and doesn't need of special hardware and complex automatic test equipment for each type of board or device.

Benefits Over Standard Test Methods

Increased complexity and lack of physical access to modern circuitry makes for costly and time-consuming testing using traditional test techniques.

Adding **testability** to a device:

- **increases** design time and costs
- **reduces** costs of design validation, manufacturing test, and system maintenance.

A structured technique such as **Boundary-Scan Testing** allows for pins-out testing to easily detect these failures.
The idea behind of **incorporate design-for-test techniques** is to:

- allows test instructions and data to be serially loaded into a device and enables the subsequent test results to be serially read out.
- allows simpler access to circuits and doesn't need of special hardware and complex automatic test equipment for each type of board or device.
- doesn't need of complex testing (and consequent loss of time).

Benefits Over Standard Test Methods

Increased complexity and lack of physical access to modern circuitry makes for costly and time-consuming testing using traditional test techniques.

Adding **testability** to a device:

- **increases** design time and costs
- **reduces** costs of design validation, manufacturing test, and system maintenance.

A structured technique such as **Boundary-Scan Testing** allows for pins-out testing to easily detect these failures.
The idea behind of **incorporate design-for-test techniques** is to:

- allows test instructions and data to be serially loaded into a device and enables the subsequent test results to be serially read out.
- allows simpler access to circuits and doesn't need of special hardware and complex automatic test equipment for each type of board or device.
- doesn't need of complex testing (and consequent loss of time).
- allows designers to add test features.

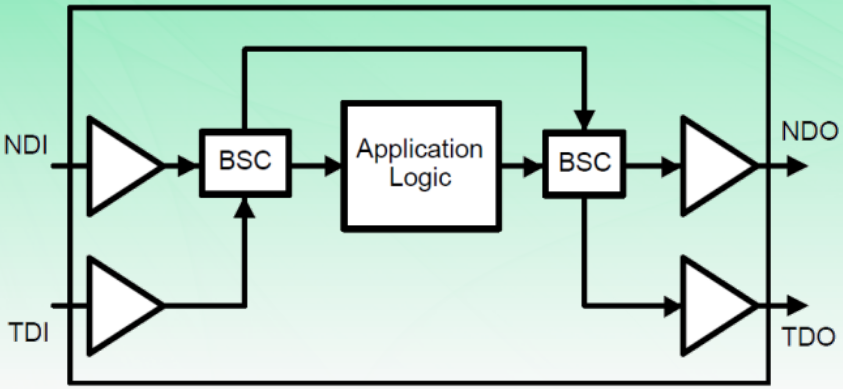
An Industry Standard: IEEE Std 1149.1-1990

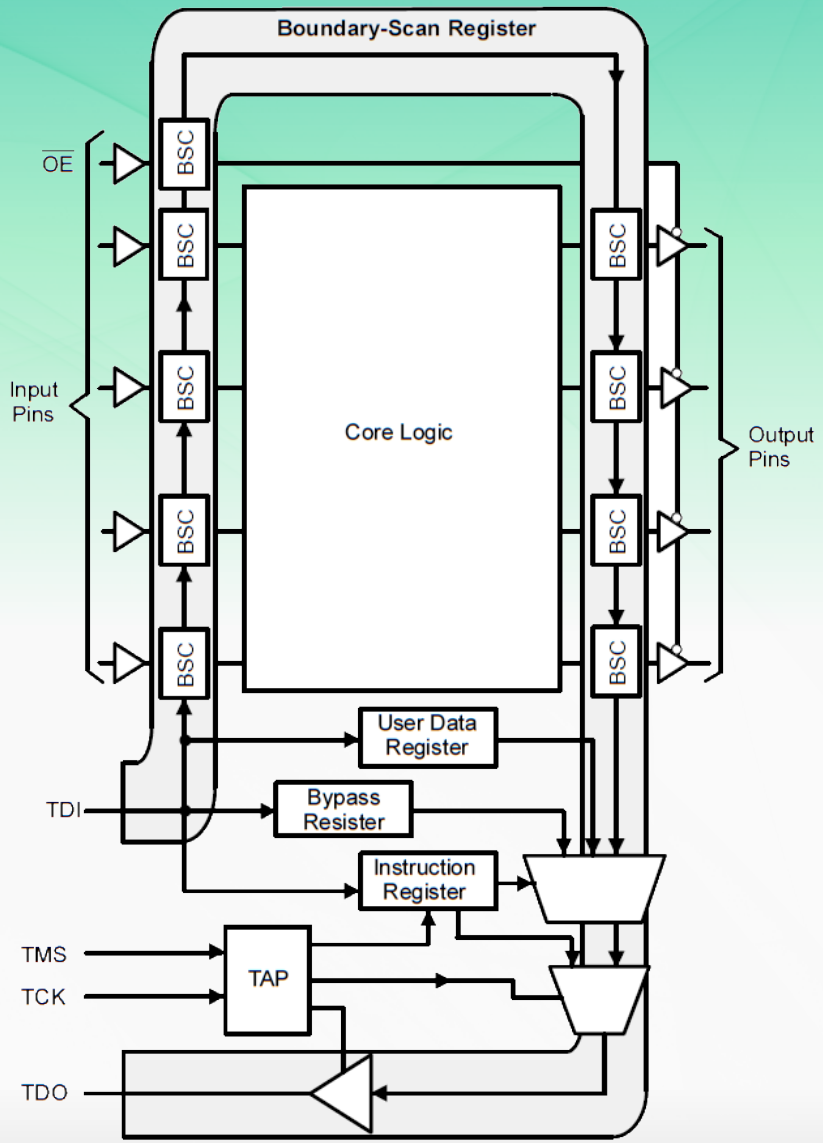
In 1985, an ad hoc group composed of key electronic manufacturers joined to form the **Joint Test Action Group (JTAG)**.

This group met to establish a solution to the problems of board test and to promote a solution as an industry standard.

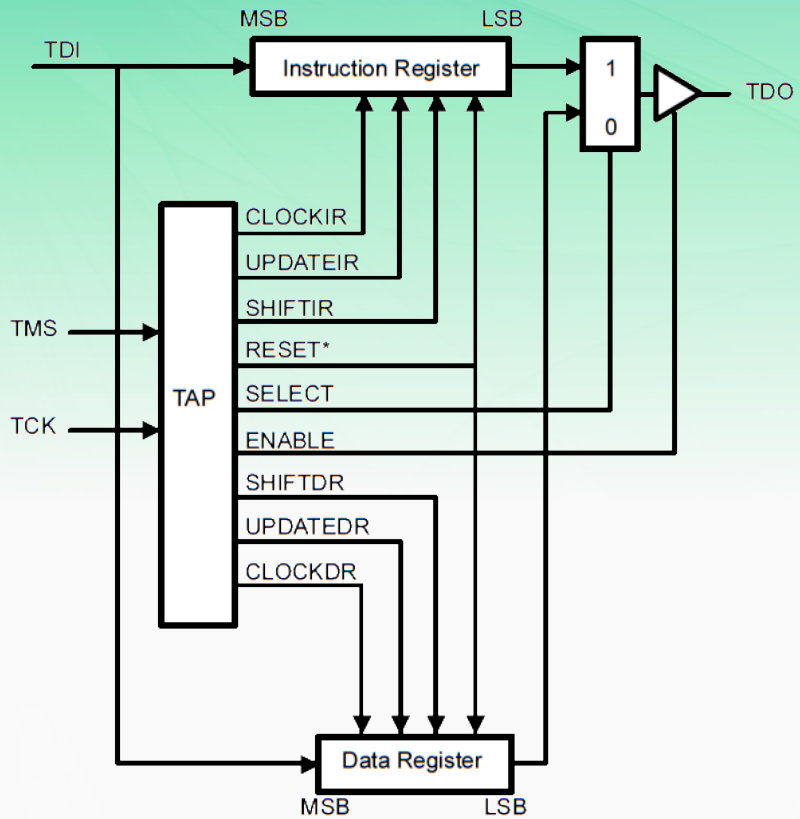
The solution, which became **IEEE Std 1149.1-1990**, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, is the basis for Texas Instruments (TIE) testability products.

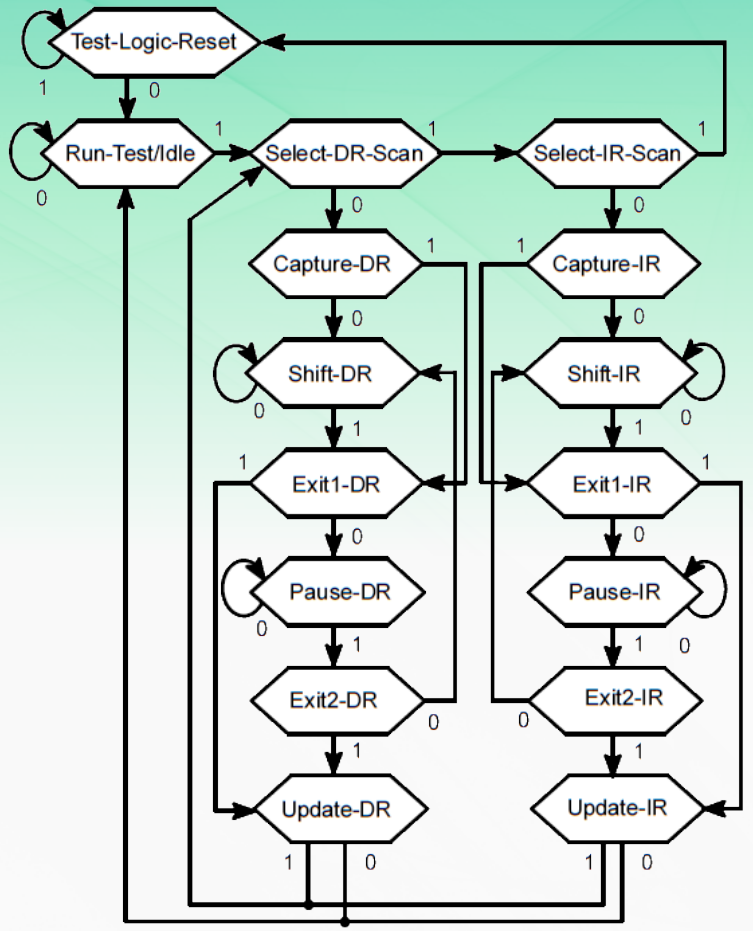
Overview and Test Interface





Test Acces Port (TAP) and State Machine





Registers

Registers

- **Instruction Register** (Required)

Registers

- **Instruction Register** (Required)
- **Data Register**

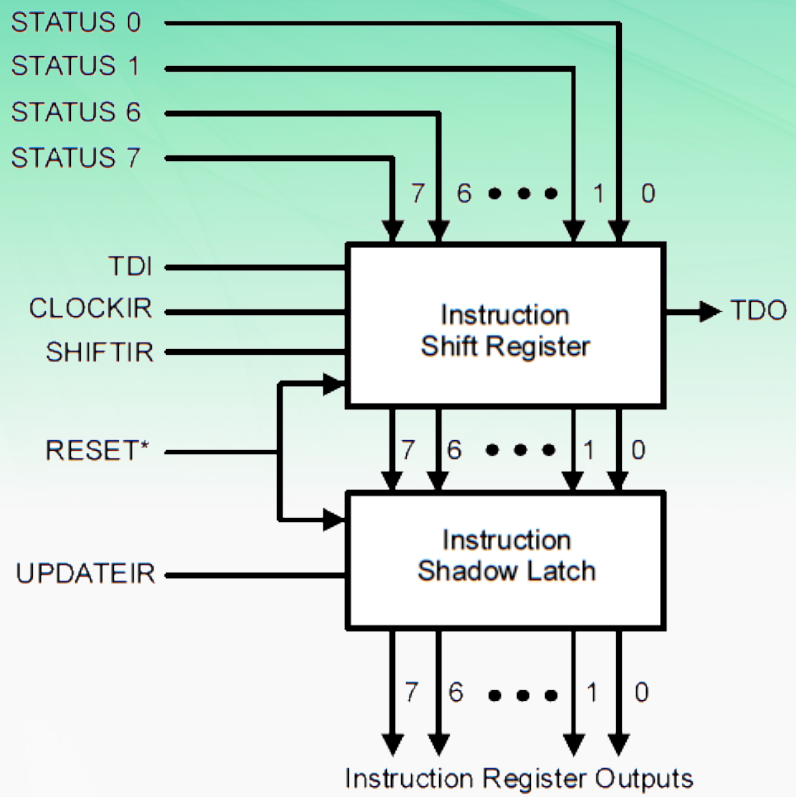
Registers

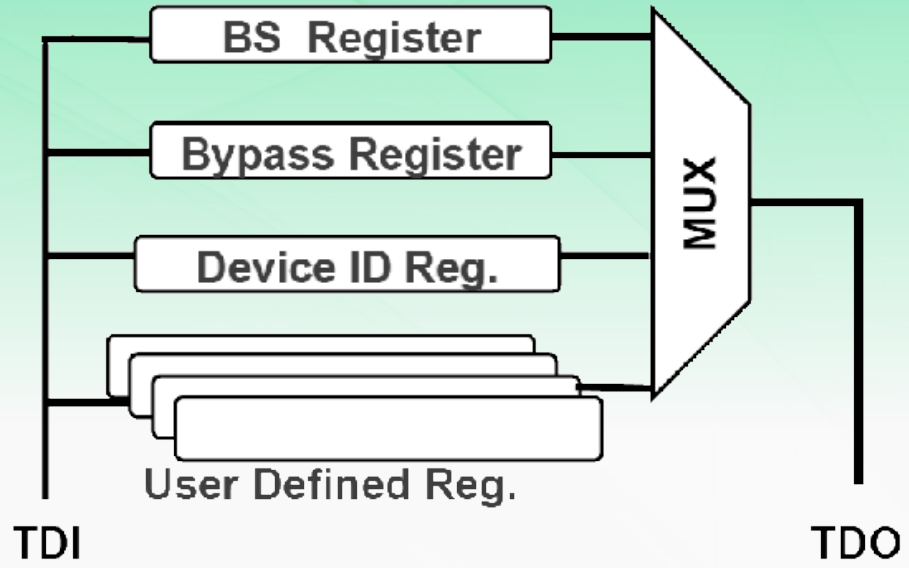
- **Instruction Register** (Required)
- **Data Register**
  - **Bypass Register** (Required)
  - **Boundary Scan Register** (Required)
  - **User Data Register** (Required)

Registers

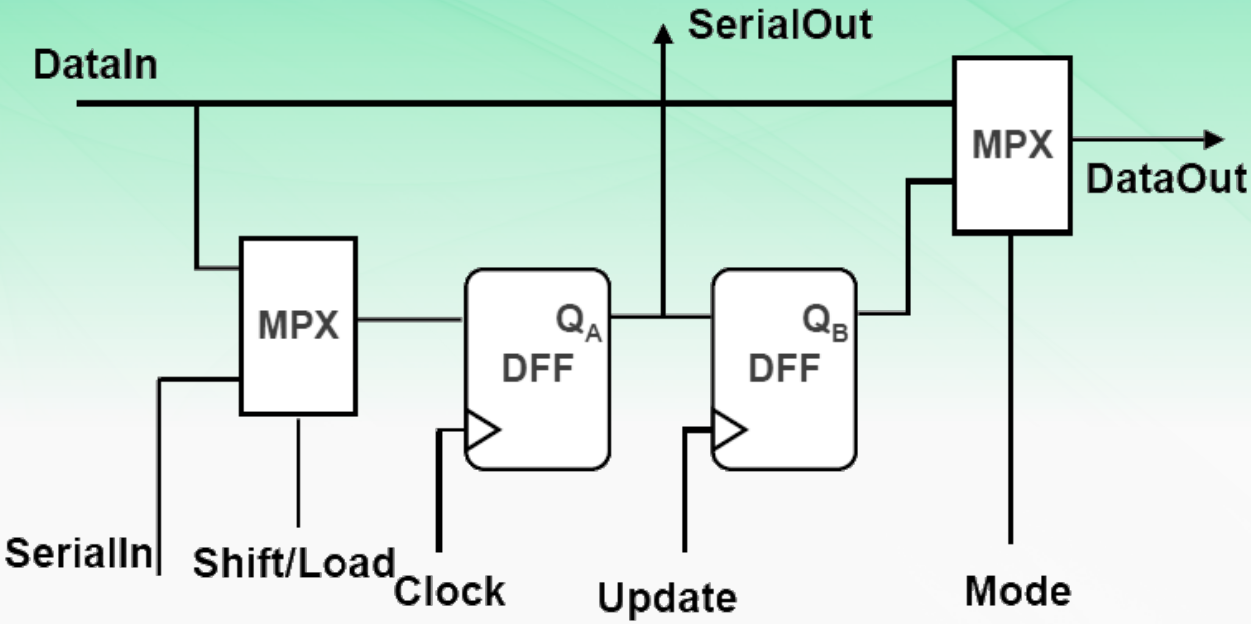
- **Instruction Register** (Required)
- **Data Register**
  - **Bypass Register** (Required)
  - **Boundary Scan Register** (Required)
  - **User Data Register** (Required)
  - Device Identification Register (Optional)
  - Others Data Register (Optional)

Instruction and Data Register Model





Boundary Scan Register (Single Cell)



Instructions

Required:

Optional:

Instructions

Required:

- **BYPASS** Instruction

Optional:

Instructions

Required:

- **BYPASS** Instruction
- **SAMPLE/PRELOAD** Instruction

Optional:

Instructions

Required:

- **BYPASS** Instruction
- **SAMPLE/PRELOAD** Instruction
- **EXTEST** Instruction

Optional:

Instructions

Required:

- **BYPASS** Instruction
- **SAMPLE/PRELOAD** Instruction
- **EXTEST** Instruction

Optional:

- INTEST Instruction
- RUNBIST Instruction
- CLAMP Instruction
- HIGHZ Instruction
- IDCODE Instruction
- USERCODE Instruction

Required Instructions

BYPASS Instruction [Bit code=all ones]: allows the IC to remain in a functional mode and selects the bypass register to be connected between TDI and TDO allowing serial data to be transferred through the IC without affecting the operation of the IC.

SAMPLE/PRELOAD Instruction [Bit code=defined by the vendor]: allows the IC to remain in its functional mode and selects the boundary-scan register to be connected between TDI and TDO. This instruction is also used to preload test data into the boundary-scan register before loading an EXTEST instruction.

EXTEST Instruction [Bit code=all zeroes]: places the IC into an external boundary-test mode and selects the boundary-scan register to be connected between TDI and TDO.

Single Event Effects

An electronic circuit, that bears no permanent hardware fault, may witness unexplained events resulting in *single bit* changes spontaneously in the system, and there is no way to repeat such failures.

The state change is a result of the *free charge* created by ionization in or close to an important node of a logic element (such as memory bit).

This phenomenon is known as a *soft fail*, to differentiate from the *hard* or *permanent fail*.

The set of all *soft* and *hard errors* are known as **Single Event Effects** (SEE).

Soft faults

A soft or non-permanent fault is a completely *random* and *non-destructive* fault and falls into two categories:

- *Transient faults*, caused by environmental conditions like temperature, humidity, pressure, voltage, power supply, vibrations, fluctuations, electromagnetic interference, ground loops, cosmic rays and α particles.
- *Intermittent faults* caused by non-environmental conditions like loose connections, aging components, critical timing, power supply noise, resistive or capacitive variations or couplings, and noise in the system.

This Work focuses on the effects of soft errors, in particular of a common SEE in the radiation environment of a HEPs experiment: the **Single Event Upsets** (SEU).

Self-correcting single-bit Hamming Code

The Hamming code is an on-line recovery mechanisms from soft-errors in order to achieve the chip robustness requirement provides a systematic procedure for generating redundant correction codes, such that there is a clear indication of any erroneous bits in the code word.

m bit word

r redundancy bits

$n = m + r$ bits of the code word

$$\boxed{2^r \geq (m + r + 1)}$$

Thus, for 16-bit data, 5 parity bits are needed:

$$\boxed{2^5 \geq 16 + 5 + 1 = \mathbf{21}}$$

Parity bits positions

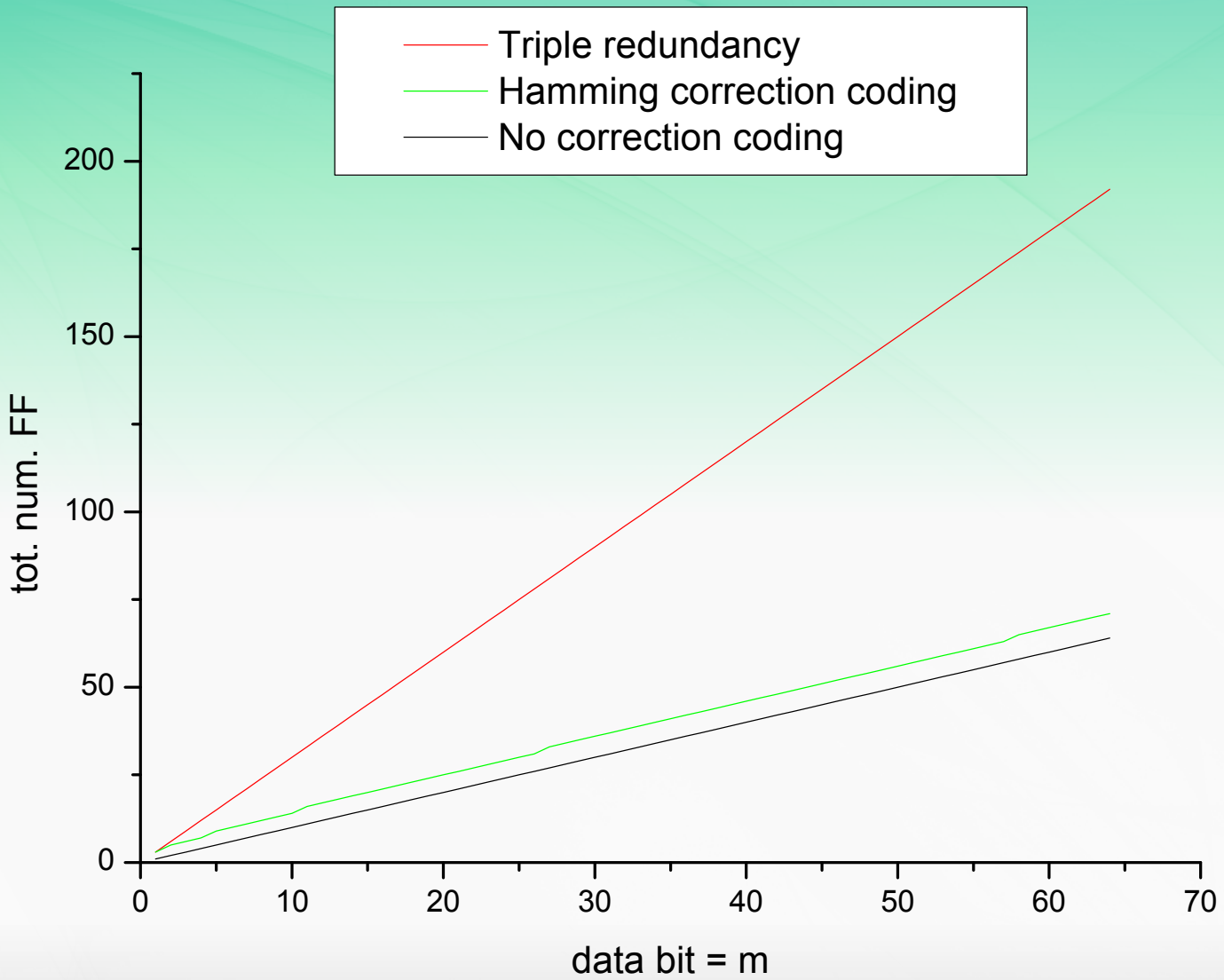
Parity bits occupy in the codeword any position whose number is a power of 2. Thus, for 16-bit data we have:

20	19	18	17	16	**15**	14	13	12	11	10	9	8	**7**	6	5	4	**3**	2	**1**	**0**
D	D	D	D	D	**P**	D	D	D	D	D	D	D	**P**	D	D	D	**P**	D	**P**	**P**
15	14	13	12	11	**4**	10	9	8	7	6	5	4	**3**	3	2	1	2	0	**1**	**0**

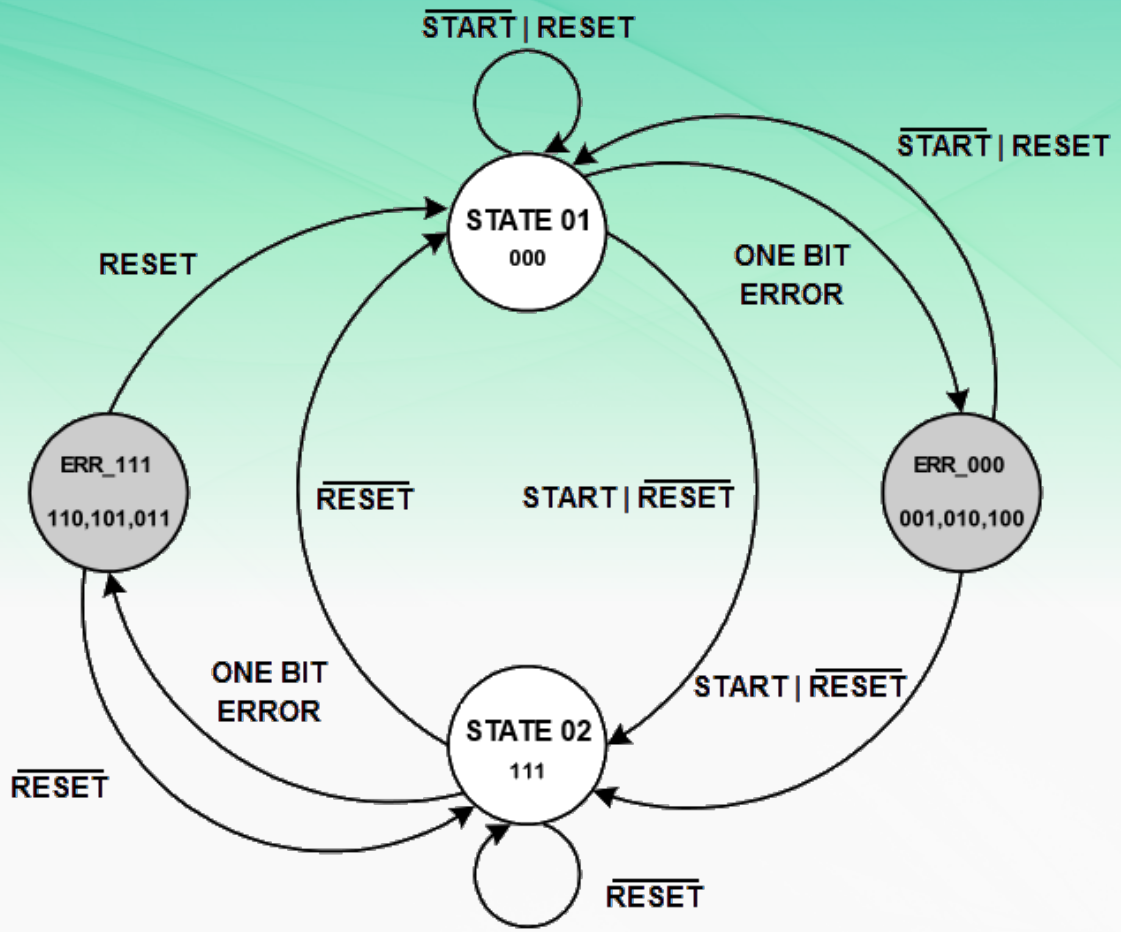
Parity bits:

$$P0 = 0 \oplus 1 \oplus 3 \oplus 4 \oplus 6 \oplus 8 \oplus 10 \oplus 11 \oplus 13 \oplus 15$$
$$P1 = 0 \oplus 2 \oplus 3 \oplus 5 \oplus 6 \oplus 9 \oplus 10 \oplus 12 \oplus 13$$
$$P2 = 1 \oplus 2 \oplus 3 \oplus 7 \oplus 8 \oplus 9 \oplus 10 \oplus 14 \oplus 15$$
$$P3 = 4 \oplus 5 \oplus 6 \oplus 7 \oplus 8 \oplus 9 \oplus 10$$
$$P4 = 11 \oplus 12 \oplus 13 \oplus 14 \oplus 15$$

Hamming code vs. triple redundancy

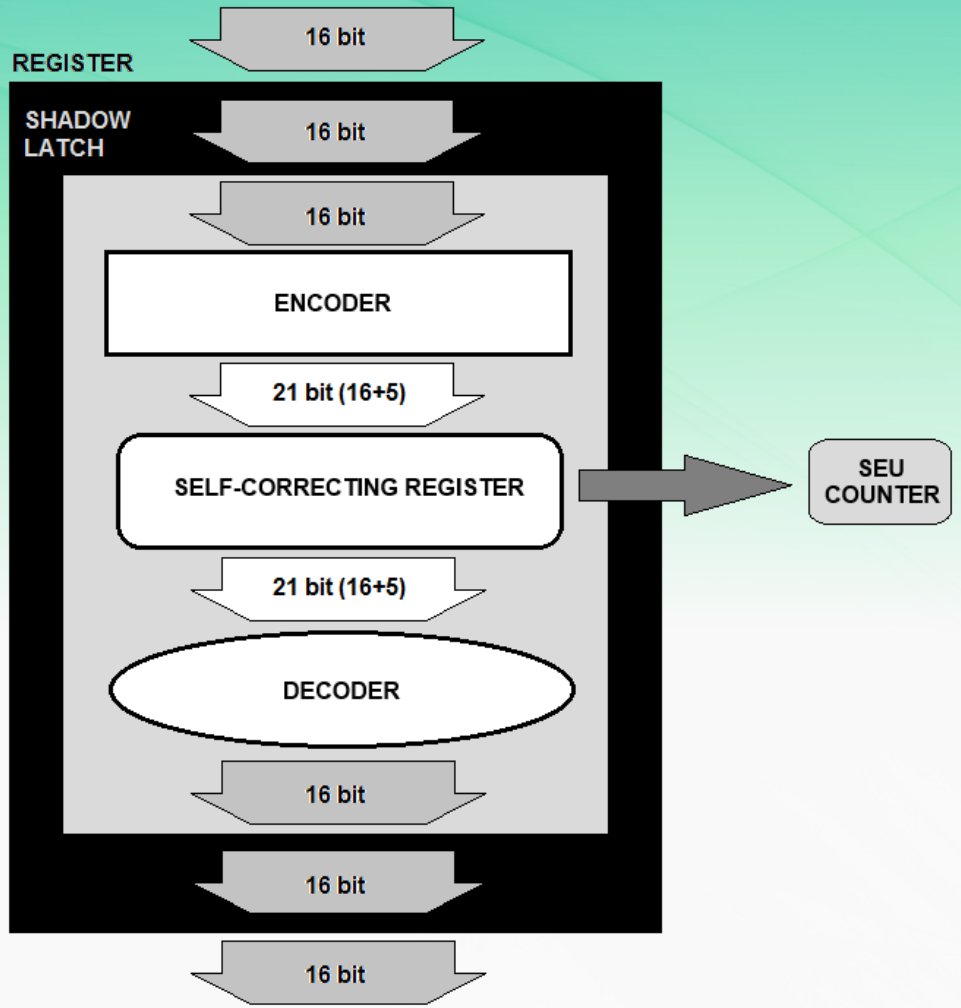


TAP controller with SEU correction



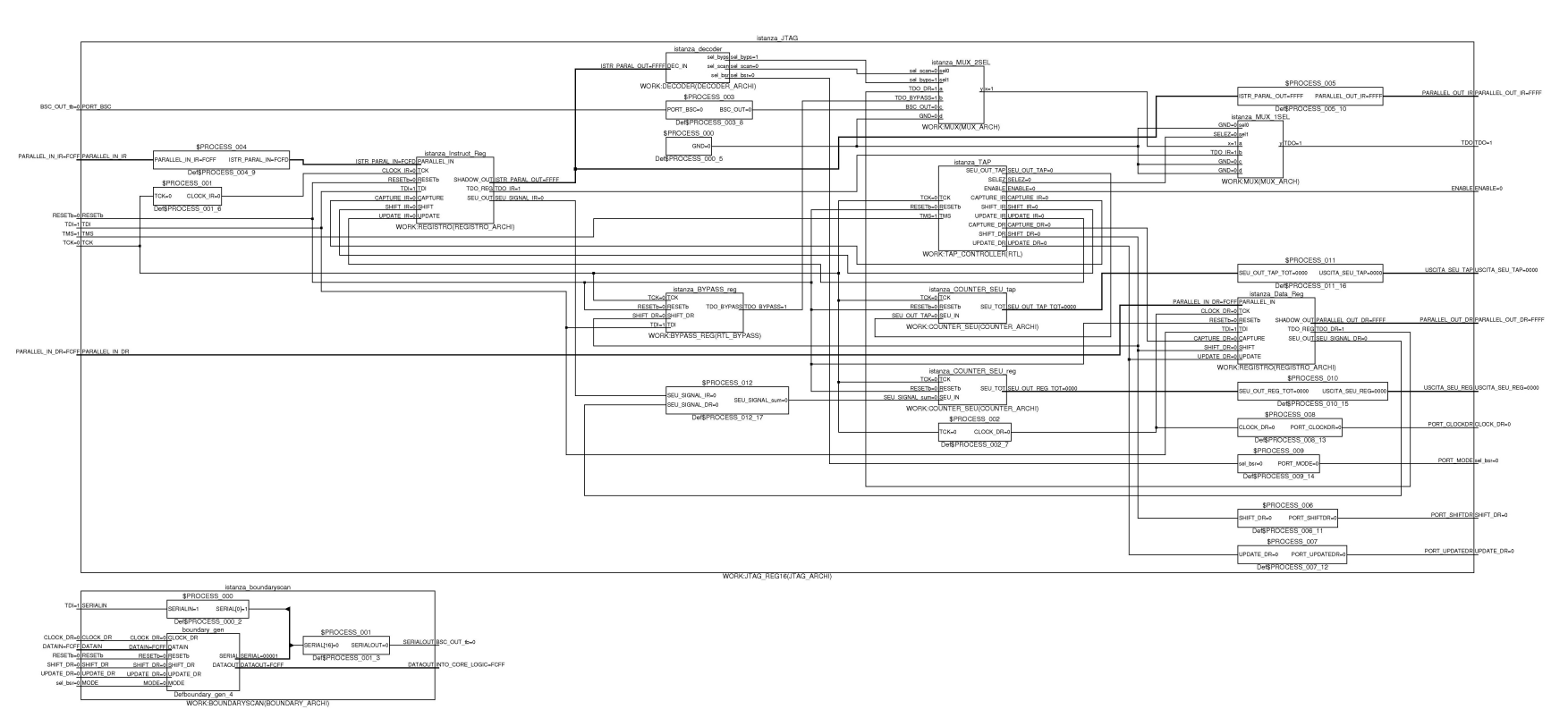
Waveform

Shadow latch with SEU correction



Waveform

JTAG simulations



JTAG

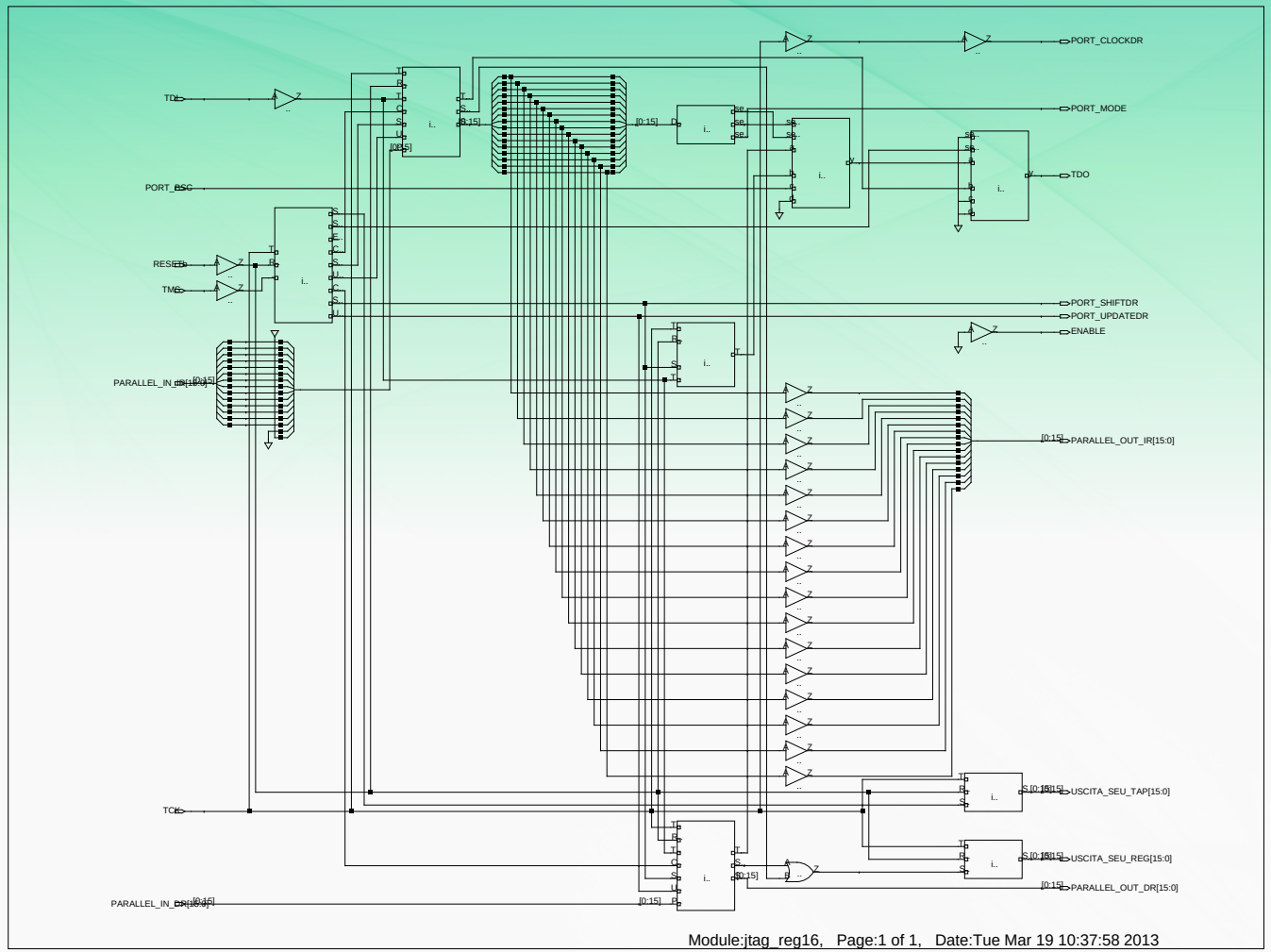
- BYPASS
- EXTEST
- SAMPLE/PRELOAD

Synthesis

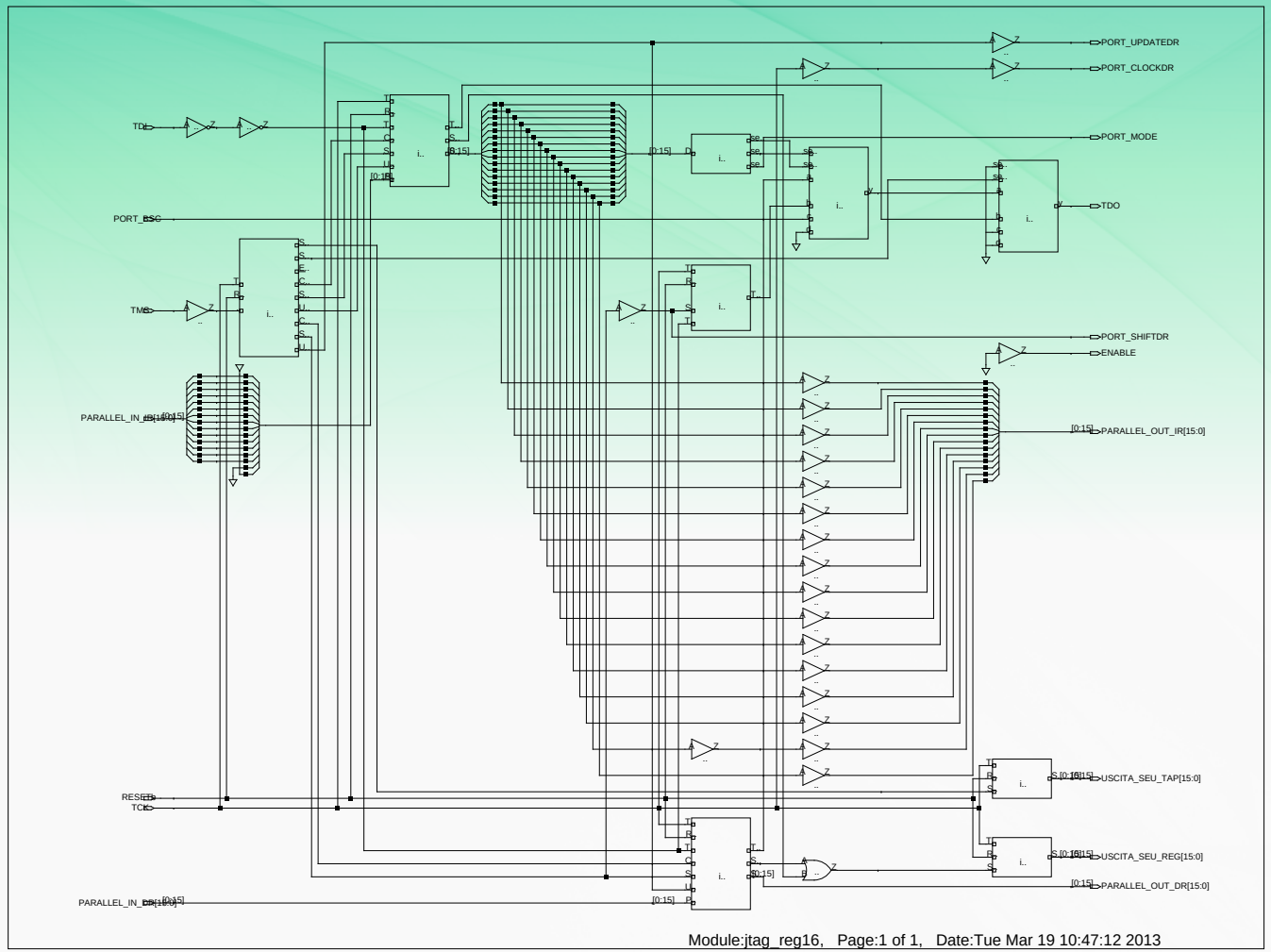
Synthesis is a process where a VHDL is compiled and mapped into an implementation technology such as an **FPGA** or an **ASIC**.

Not all constructs in VHDL are suitable for synthesis. While different synthesis tools have different capabilities, there exists a common synthesizable subset of VHDL that defines what language constructs and idioms map into common hardware for many synthesis tools. IEEE 1076.6 defines a subset of the language that is considered the official synthesis subset.

Synthesis: IBM 0.13 μm



Synthesis: UMC 0.11 μm



Area reports (IBM 100 MHz)

	Inst.	Inst. Ham	Area (μm)	Area Ham. (μm)	Area Ham. %	Area %	ΔA (μm)	ΔA %
seq.	74	157	3144.960	6218.880	75.8	47.0	3073.92	97.7
logic	90	587	1004.16	7023.36	24.2	53.0	6019.2	599.4
tot	164	744	4149.120	13242.240	100.0	100.0	9093.12	219.6

Power reports (IBM 100 MHz)

	Inst.	Inst. Ham.	Leak. (μW)	Leak. Ham. (μW)	Leak. %	Leak. Ham. %	ΔP_L (μW)	ΔP_L %
seq.	74	157	73.597	149.320	88.9	72.4	75.723	102.9
logic	90	587	9.204	56.918	11.1	27.6	47.714	518.4
tot	164	744	82.802	206.238	100.0	100.0	123.437	149.1

Frequency	Cells	Leakage Power (μW)	Dynamic Power (μW)	Total Power (μW)
100 MHz	164	82.807	204.788	287.589
100 MHz (Ham)	744	206.238	707.648	913.886
Δ	580	123.437	502.860	626.297

Area reports (UMC 100 MHz)

	Inst.	Inst. Ham	Area (μm)	Area Ham. (μm)	Area Ham. %	Area %	ΔA (μm)	ΔA %
seq.	69	148	1947.840	3997.440	60.0	43.9	2049.6	105.2
logic	143	597	1300.8	5118.72	40.0	56.2	3817.92	293.5
tot	212	745	3248.640	9116.160	100.0	100.0	5867.52	180.6

Power reports (UMC 100 MHz)

	Inst.	Inst. Ham.	Leak. (μW)	Leak. Ham. (μW)	Leak. %	Leak. Ham. %	ΔP_L (μW)	ΔP_L %
seq.	69	148	0.107	0.223	33.8	27.5	0.116	108.4
logic	143	597	0.210	0.587	66.2	72.5	0.377	192.0
tot	212	745	0.317	0.811	100.0	100.0	0.493	155.8

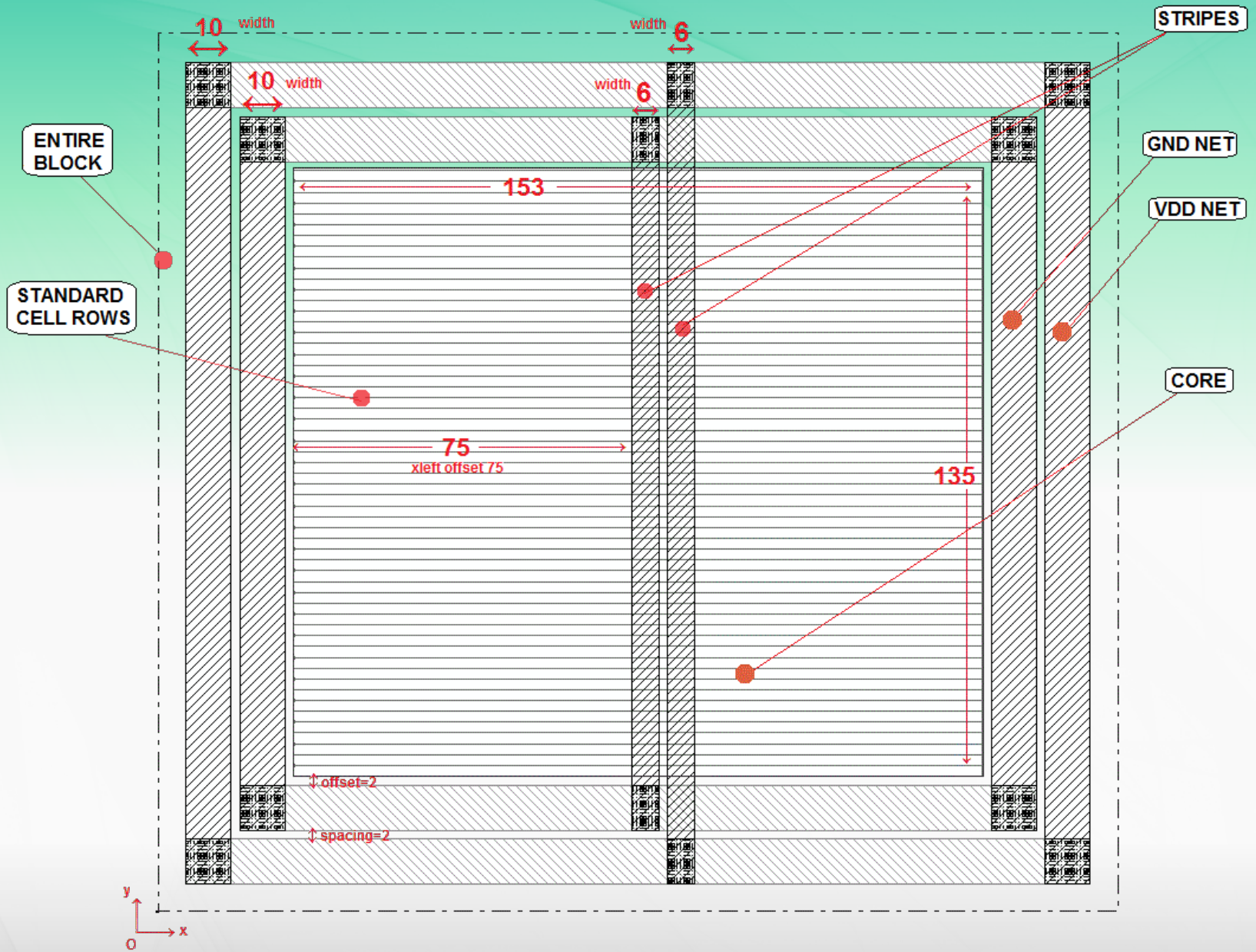
Frequency	Cells	Leakage Power (μW)	Dynamic Power (μW)	Total Power (μW)
100 MHz	212	0.317	279.226	279.544
100 MHz (Ham)	745	0.811	727.912	728.723
Δ	533	0.493	448.686	449.179

Place and Route

Place and route is the final stage in the design of printed circuit boards, integrated circuits, and field-programmable gate arrays.

- **Placement** involves deciding where to place all electronic components, circuitry, and logic elements in a generally limited amount of space in order to minimize the lenght of the interconnections with special emphasis on the timing critical paths.
- **Routing** indeed decides the exact design of all the wires needed to connect the placed components. This step must implement all the desired connections while following the rules and limitations of the manufacturing process.

Floorplan


10 width
10 width
width 6
width 6
STRIPES
ENTIRE BLOCK
GND NET
VDD NET
153
STANDARD CELL ROWS
CORE
75
xleft offset 75
135
offset=2
spacing=2
y
x
O

Post Route Timing Reports Summary for IBM and UMC at 100 MHz

		all	reg2reg	in2reg	reg2out	in2out
IBM	WNS (ns):	0.071	0.071	1.635	0.345	0.205
UMC		0.194	0.194	1.013	0.411	0.322
IBM	TNS (ns):	0.000	0.000	0.000	0.000	0.000
UMC		0.000	0.000	0.000	0.000	0.000
IBM	Violating Paths:	0	0	0	0	0
UMC		0	0	0	0	0
IBM	All Paths:	293	184	56	68	2
UMC		481	264	203	68	2

Slack

Post Route Timing Reports Summary for IBM and UMC at 500 MHz

		all	reg2reg	in2reg	reg2out	in2out
IBM	WNS (ns):	0.068	0.068	1.538	0.359	0.198
UMC		0.176	0.176	1.337	0.388	0.331
IBM	TNS (ns):	0.000	0.000	0.000	0.000	0.000
UMC		0.000	0.000	0.000	0.000	0.000
IBM	Violating Paths:	0	0	0	0	0
UMC		0	0	0	0	0
IBM	All Paths:	309	200	56	68	2
UMC		453	236	187	68	2

Slack

Final PnR Comparison

Density:

For both processes, both frequencies (100 MHz and 500 MHz) are smoothly reached.

	100 MHz	500 MHz
IBM	69.476 %	93.223 %
UMC	47.866 %	92.743 %
Difference	**21.610** %	**0.480** %

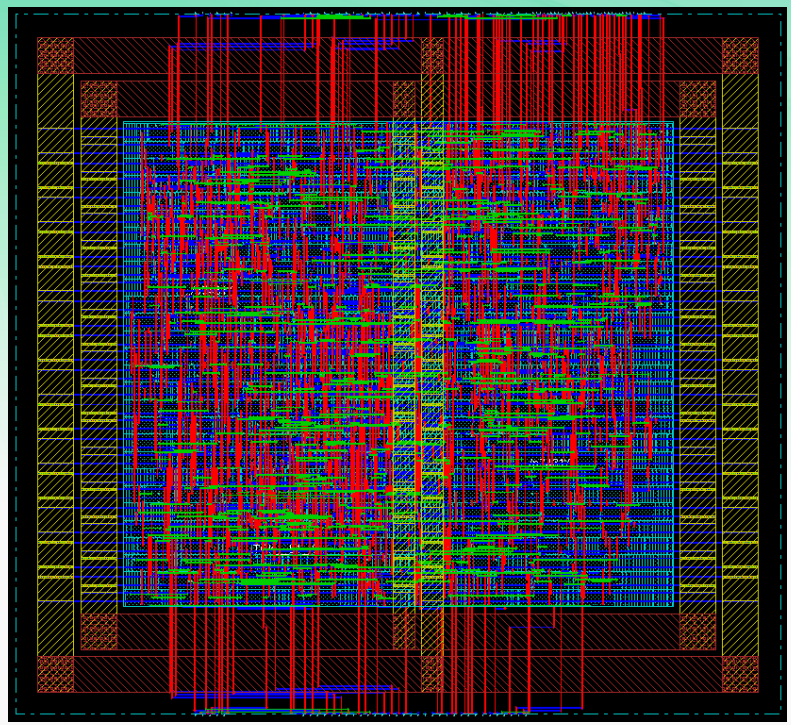
What this means?

- **IBM** technology is optimized for **high frequencies**;
- **UMC** technology is optimized for the **area**.

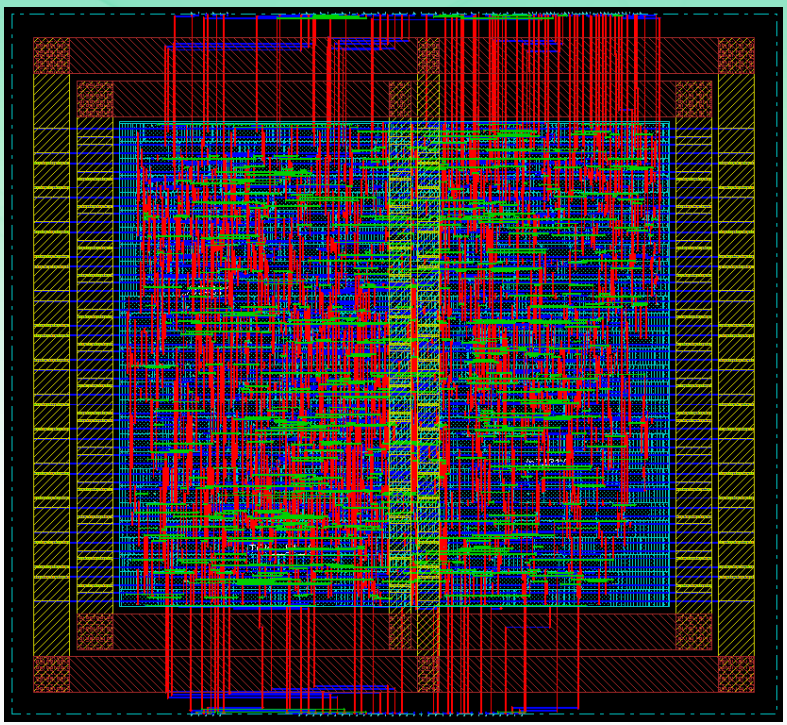
Thus:

- at low frequencies the density for IBM is rather higher than the density of the UMC;
- at high speeds they become almost equal (UMC requires a greater number of ports - and thus area - to work properly.)

Final Layout at 100 MHz

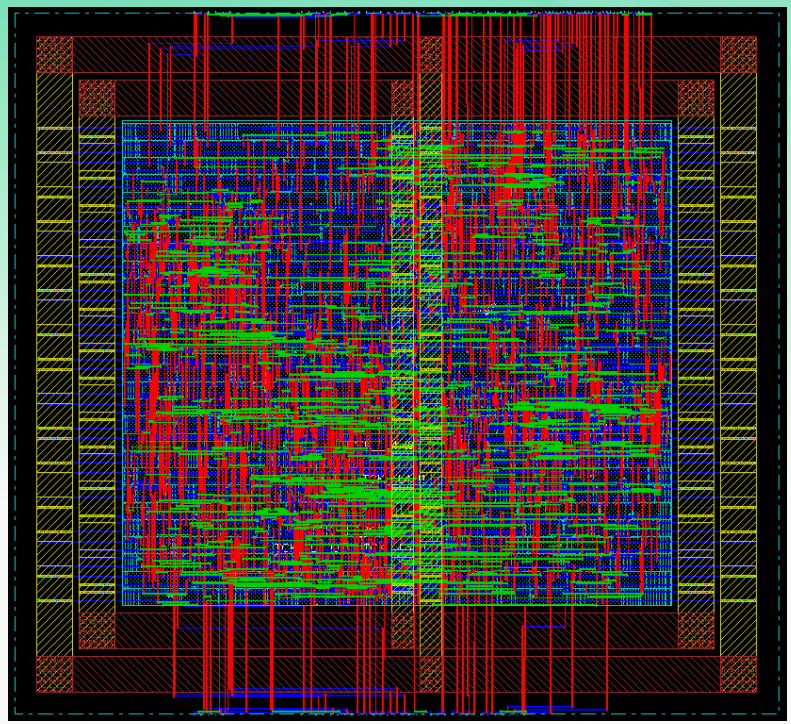


(a) Final layout for IBM 0.13 μm at 100 MHz.

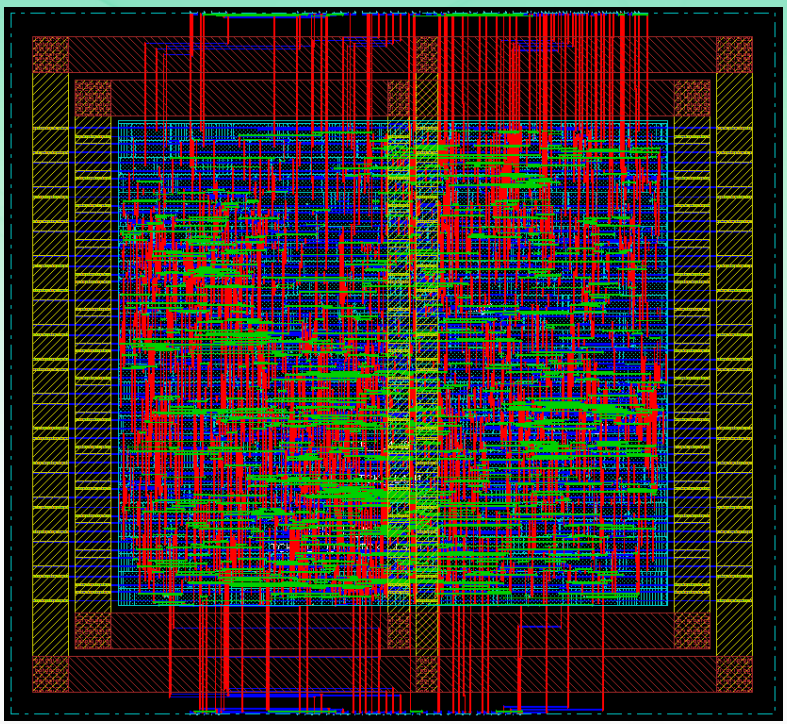


(b) Final layout for UMC 0.11 μm at 100 MHz.

Final Layout at 500 MHz



(c) Final layout for IBM 0.13 μm at 500 MHz.



(d) Final layout for UMC 0.11 μm at 500 MHz.

Thank you for your attention!