Asynchronous Circuits Design

Marc Renaudin

TIMA

Marc.Renaudin@imag.fr
http://tima.imag.fr
http://tima.imag.fr/cis/

Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
- Design experiments
- Conclusion and prospects

CIS group of TIMA laboratory

TIMA (http://tima.imag.fr)

About 130 people

Six research groups

- MNS : Micro and Nano Systems
- RMS : Reliable Mixed Signal Systems
- SLS : System Level Synthesis
- VDS : Verification and modeling of Digital Systems
- QLF : QuaLiFication of circuits
- CIS : Concurrent Integrated Systems

CIS group of TIMA laboratory

"CIS" Group (http://tima.imag.fr/cis)

About 25 people

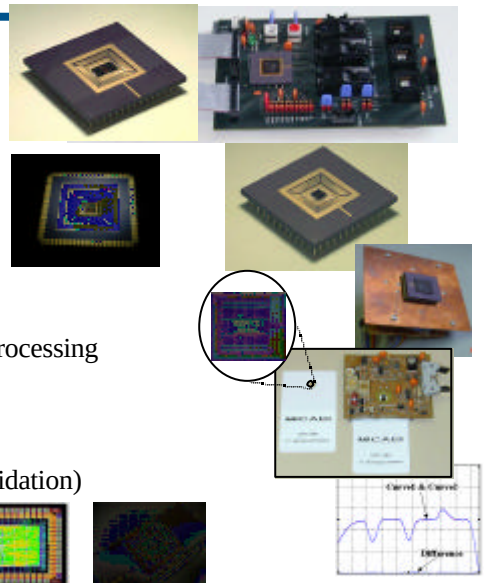
Research topics

- *Asynchronous circuits design and prototyping*
- *CAD Tools for Asynchronous circuits and systems*
- *Formal verification of asynchronous designs (Coll. with VDS D. Borrione)*
- *Hardware-software design for low power low noise*
- *Secure circuit design*
- *SoCs and NoCs design*
- *Signal driven sampling and processing*

CIS group of TIMA laboratory

Main results (http://tima.imag.fr/cis)

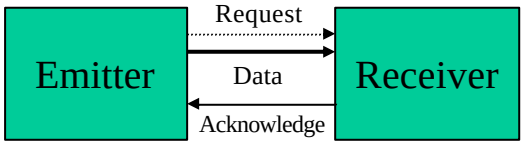
- Asynchronous Processors
  - ASPRO (16 bit RISC)
  - MICA (8 bit CISC)
- Contactless Smart Card
- Secure chip design (DES, AES)
- A-ADC & non-uniformly sampled signal processing
- High dynamic range CMOS image sensors
- TAST framework (modeling, synthesis, validation)



Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
- Design experiments
- Conclusion and prospects

Asynchronous circuit design principles



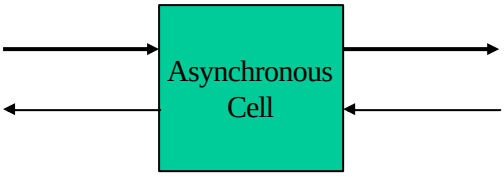
- Unified asynchronous interface
- Control and data are encoded (together or not)
- Two basic rules :
  - the emitter issues a request when a data is valid
  - the receiver issues an acknowledge when the data is processed

→ **Several hardware implementation of the handshake protocol**

→ **Delay insensitivity**

Asynchronous circuit design principles

Features of a basic asynchronous cell



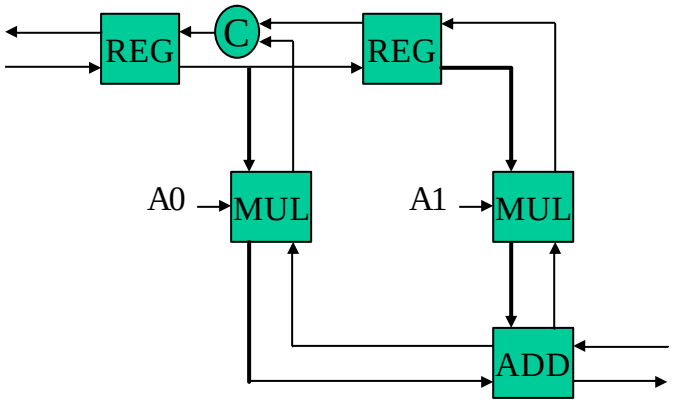
- Bit level
- Arithmetic function
- Complex function

- Maximum speed : minimum forward latency
- Maximum throughput : minimum cycle time
- Respect the handshake protocol

→ **design issues solved locally => cells are easy to reuse**

Asynchronous circuit design principles

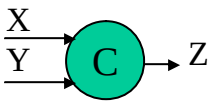
Design of a FIR filter

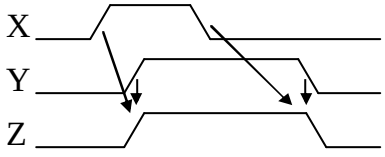


Asynchronous circuit design principles

The C-Element or Muller gate

Symbol

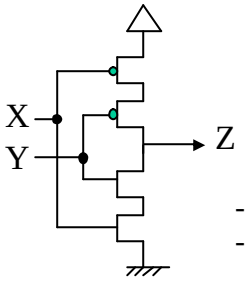




Truth table

X	Y	Z
0	0	0
0	1	Z^{-1}
1	0	Z^{-1}
1	1	1

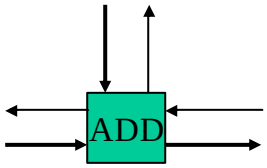
$Z = XY + Z(X+Y)$



- State holding
- Reset

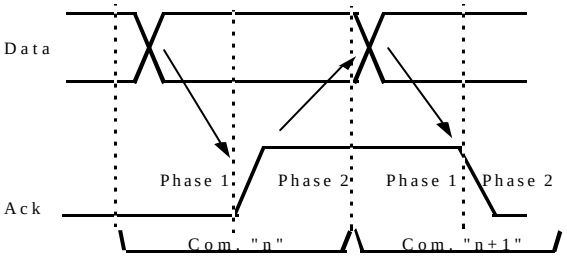
Asynchronous circuit design principles

- Protocol
  - Two phases
  - Four phases
- Signaling
  - Data encoding / Request
    - Three states
    - Four states
  - Acknowledge



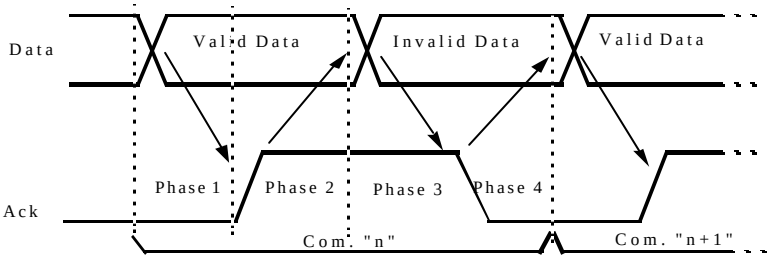
Asynchronous circuit design principles

Two Phase Protocol



Asynchronous circuit design principles

Four Phase Protocol

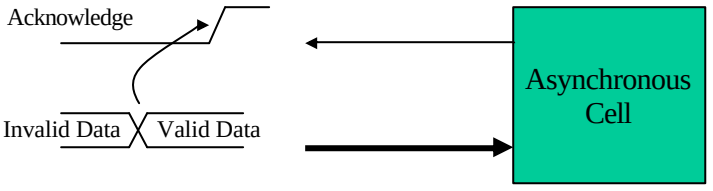


=> Several derivatives exist

Asynchronous circuit design principles

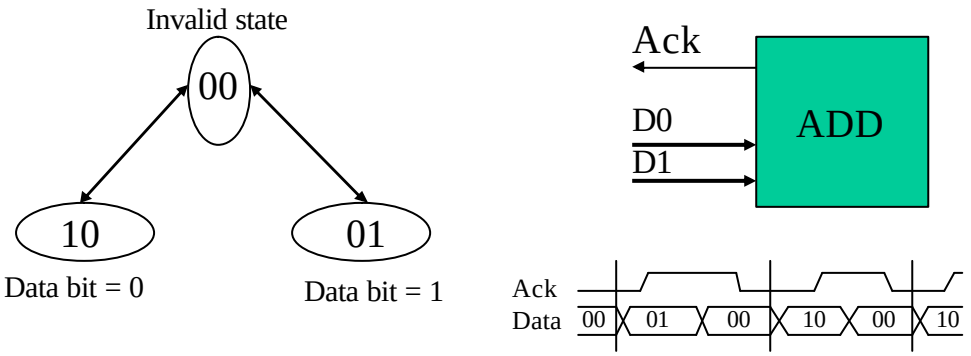
Data Valid/Invalid Signaling

- Three state coding
- Four state coding



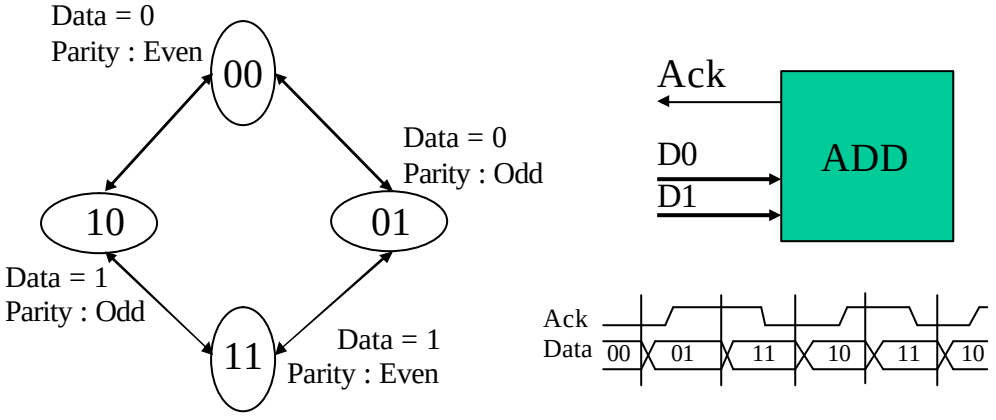
Asynchronous circuit design principles

Data encoding : Three states (dual rail)



Asynchronous circuit design principles

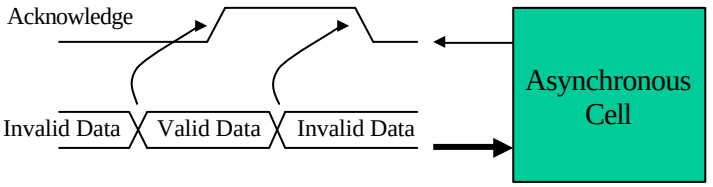
Data encoding : Four states (dual rail)



Asynchronous circuit design principles

Completion Signal Generation

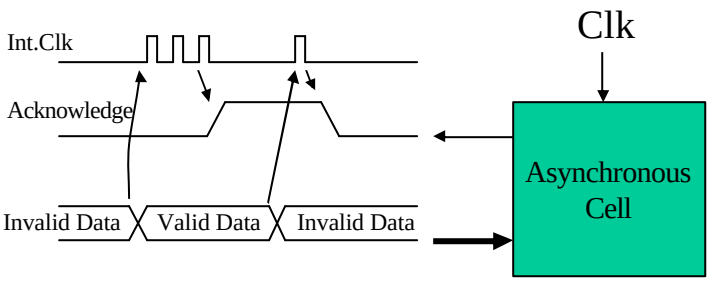
- Internal clock
- Use of a delay model
- Current sensing
- Use the data encoding



Asynchronous circuit design principles

Completion Signal Generation

- Internal clock

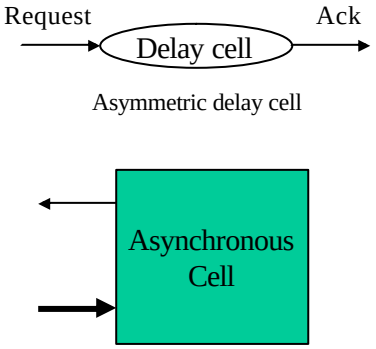


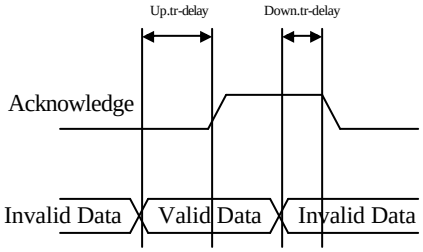
(First Data Flow Computer DDM1, 1978)

Asynchronous circuit design principles

Completion Signal Generation

- Use of a delay model

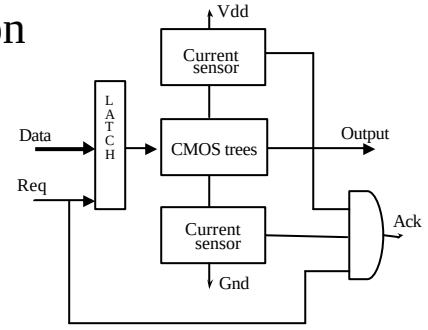


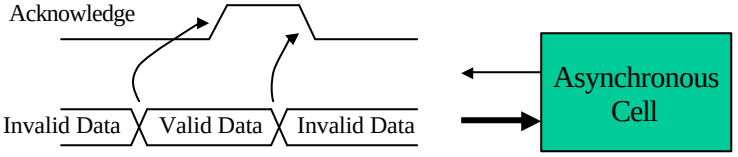


Asynchronous circuit design principles

Completion Signal Generation

- Current sensing



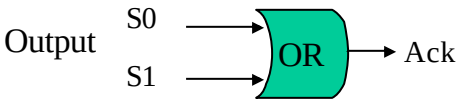


Asynchronous circuit design principles

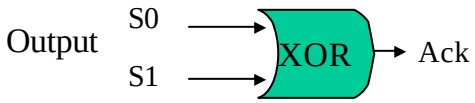
Completion Signal Generation

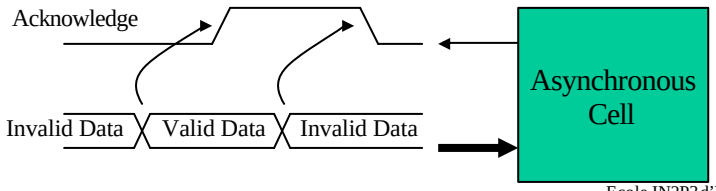
- Using data encoding (dual rail)

Three state encoding



Four state encoding

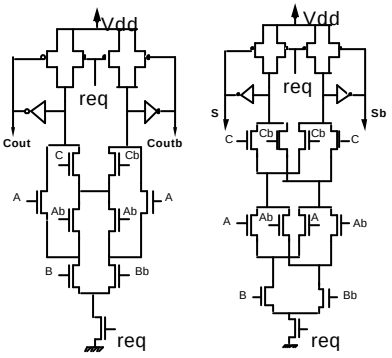


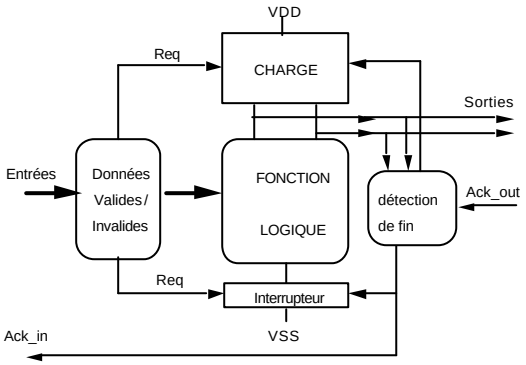


Asynchronous circuit design principles

Completion Signal Generation

- Using data encoding





Asynchronous circuit design principles

- Conclusion
  - Asynchronous circuit communicate using an handshaking protocol
  - Data/Request have to be encoded
  - A completion signal is required

→ The implementation may be delay insensitive

→ Hazard free logic is required

→ Hardware overhead ?

Asynchronous Circuits Design

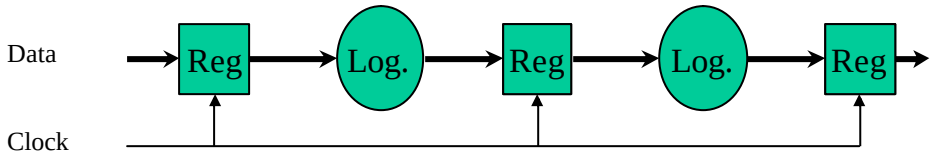
- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties
- Design experiments and products
- Conclusion and prospects

Asynchronous circuit classes

- Hazard free logic
- QDI / Speed Independent Circuits
  - Data path : a dual-rail OR Gate
  - Sequencing : the Q-Element
- Bounded delay circuits
  - Huffman circuits / Burst mode circuits
  - Sequencing : the Q-Element
- Micropipeline circuits

Asynchronous circuit classes

- Synchronous circuits



  - Time is discrete

→ combinatorial logic is simple (hazards ignored)

→ trivial communication mechanism

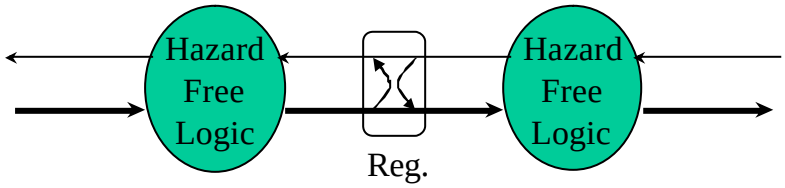
→ worst case approach

→ **Global clock => Global timing assumption**

Asynchronous circuit classes

- Asynchronous circuits

No global clock = no global timing assumption
Sequencing is based on Handshaking
=> Hazard free logic is required

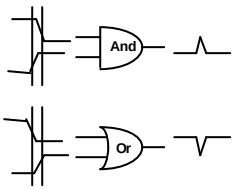


- Delay insensitive circuits
- Quasi delay insensitive circuits
- Speed independent circuits
- Huffman / Burst-mode circuits
- Micropipeline

Robustness & Complexity are decreasing : more timing assumptions

Hazards

- Static hazards
- Dynamic hazards
- Combinatorial hazards
- Functional hazards
- Sequential hazards

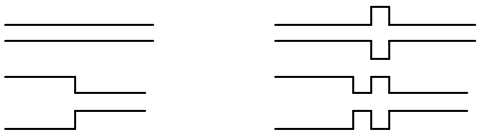


Hazards : introduction

- Hazard = spurious signal change
- Hazards appear during processing
  - it is then related to signal dynamics and component delays (wires and gates)

- Avoiding hazard implies :
  - to characterize the interaction between the circuit and its environment (define assumptions)
  - to characterize wire and gate delays (define assumptions)

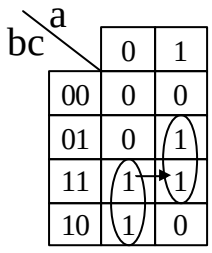
- Static hazard 0
- Static hazard 1
- Dynamic hazard 0
- Dynamic hazard 1



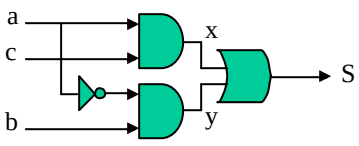
No hazard With hazard

Hazards : Single Input Change (SIC)

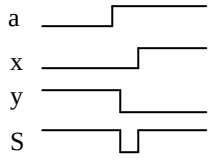
Kmap to circuit with hazard



S = ac + /ab



Static hazard 1

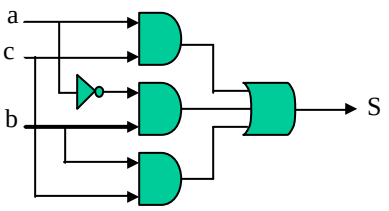


Hazards : Single Input Change (SIC)

Solution

$S = ac + /ab + \mathbf{bc}$

bc \ a	0	1
00	0	0
01	0	1
11	1	1
10	1	0



Hazards : Single Input Change (SIC)

Result

It has been proven that no hazard are generated during 0 to 0, 0 to 1 and 1 to 0 transitions in an AndOr circuit.

Dynamic hazard cannot be generated with AndOr circuits under the SIC condition

It is then only required to suppress 1 to 1 static hazards !

=> The solution is to cover 1 to 1 transitions in the Karnaugh Map

Hazards : Multiple Input Change (MIC)

Functional, Static and Dynamic hazards can be generated in this case !

Functional hazard : inherent to the specification. There is no possible hazard free implementation (with no timing assumption). The specification must be free of functional hazard.

cd \ ab	00	01	11	10
00	1	1	1	1
01	0	1	1	0
11	0	1	1	0
10	1	1	0	0

0010 to 0111 transition.

If b changes after d there is a static 0 hazard at the output !

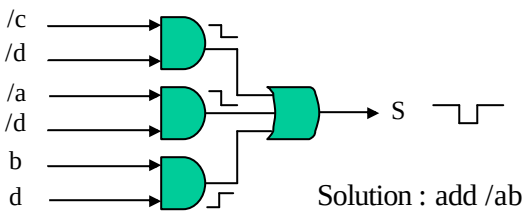
Hazards : Multiple Input Change (MIC)

Logic hazard : static 1

$$S = /c/d + /a/d + bd$$

cd \ ab	00	01	11	10
00	1	1	1	1
01	0	1	1	0
11	0	1	1	0
10	1	1	0	0

0100 to 0111 transition.



Solution : add /ab

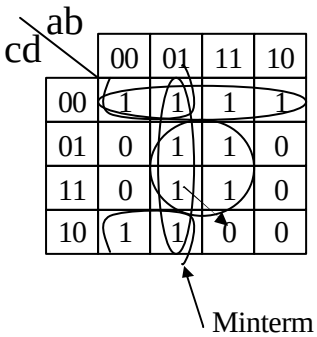
No static 0 hazard in AndOr circuit under the MIC condition !

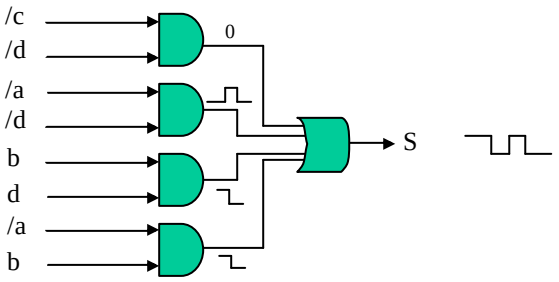
Hazards : Multiple Input Change (MIC)

Dynamic hazard : dynamic 0

$S = /c/d + /a/d + bd + /ab$

0111 to 1110 transition. No functional hazard.



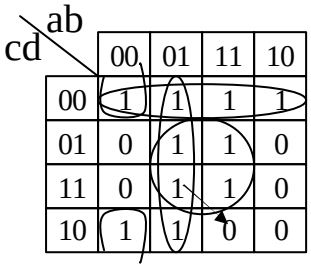


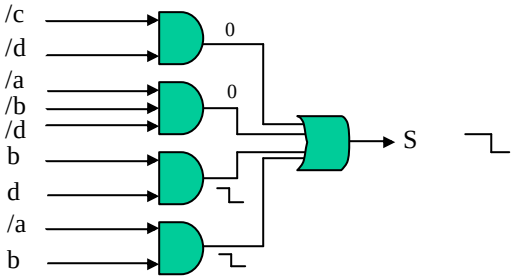
Minterm /a/d is activated and deactivated !

Hazards : Multiple Input Change (MIC)

Dynamic hazard : dynamic 0

Solution : strengthen /a/d ⇒ $S = /c/d + /a/d/b + bd + /ab$





Hazards : conclusion

Avoiding hazard is a Karnaugh Map covering problem !

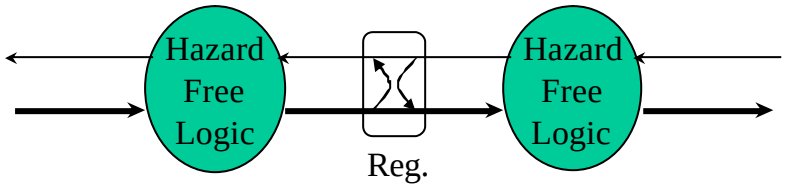
1 to 1 : the transition must be covered
1 to 0 and 0 to 1 : avoid activation and deactivation of a minterm
0 to 0 : no hazard in an AndOr circuit.

The covering problem may not have a solution if there are several MIC transitions. Therefore a delay insensitive implementation may no exist !

Asynchronous circuit classes

- Asynchronous circuits

No global clock = no global timing assumption
Sequencing is based on Handshaking
=> Hazard free logic is required



- Delay insensitive circuits
- Quasi delay insensitive circuits
- Speed independent circuits
- Huffman / Burst-mode circuits
- Micropipeline

Robustness & Complexity

are decreasing :

more timing assumptions

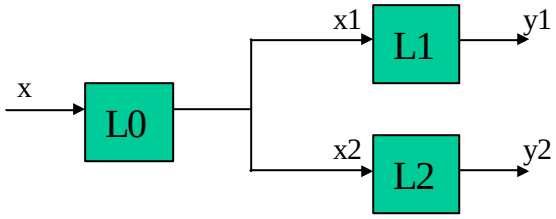
Asynchronous circuit classes

- Hazard free logic
- QDI / Speed Independent Circuits
  - Data path : a dual-rail OR Gate
  - Sequencing : the Q-Element
- Bounded delay circuits
  - Huffman circuits / Burst mode circuits
  - Sequencing : the Q-Element
- Micropipeline circuits

Asynchronous circuit classes

QDI Asynchronous circuits

- Functionally correct without any assumption on the wire and gate delays (unbounded delay model) except...
- "Isochronic fork" timing assumption



→ High level of robustness (with respect to delay variations)

Asynchronous circuit classes

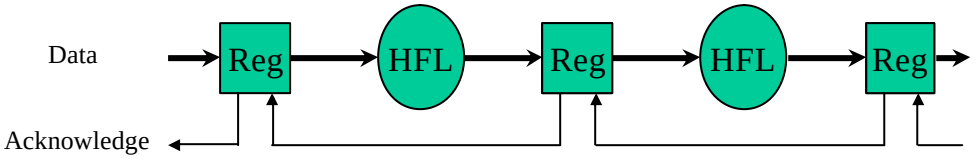
Speed Independent Asynchronous circuits

- Functionally correct whatever the delays in the gates (unbounded delay model)
- The wires are assumed to be zero delay
- => all wires are required to respect the isochronic fork property

→ Less accurate than the QDI model

QDI/SI asynchronous circuits

- Quasi Delay Insensitive & Speed Independent :
  => hazard free control logic & hazard free data-paths



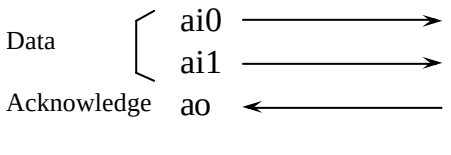
– time is no longer discrete
→ hazard free combinatorial logic
→ handshake based communications
→ mean time approach

→ **No timing assumption**

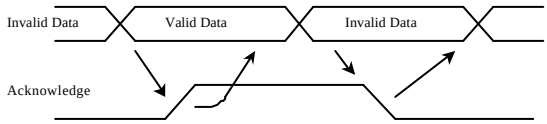
QDI/SI asynchronous circuits

- Implementing delay insensitivity : examples
  - Choice of a communication protocol (request - acknowledge)
    - 1 bit Channel
      - Data encoding

Data	ai0	ai1
0	1	0
1	0	1
Invalid	0	0

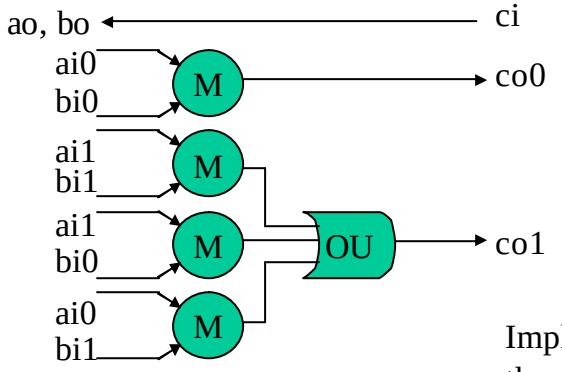


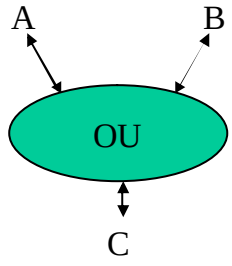
      - 4 phase protocol



QDI/SI asynchronous circuits

- An example : dual-rail OR Gate

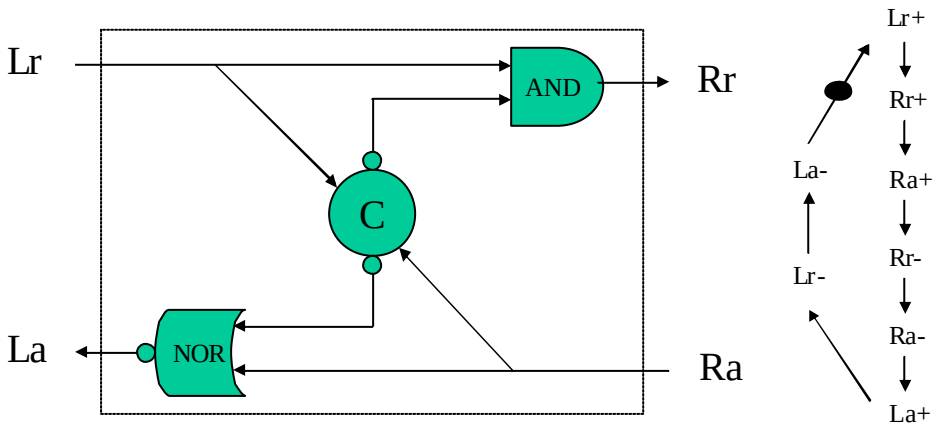




Implement both the function and the protocol

QDI/SI Asynchronous circuits

- An example : the Q-Element

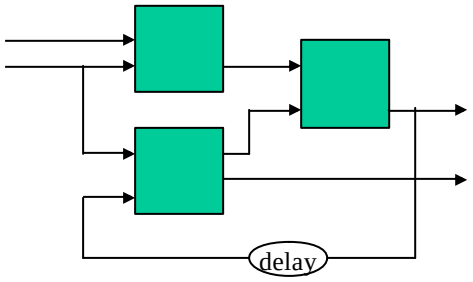


Asynchronous circuit classes

- Hazard free logic
- QDI / Speed Independent Circuits
  - Data path : a simple OR Gate
  - Sequencing : the Q-Element
- Bounded delay circuits
  - Huffman circuits / Burst mode circuits
  - Sequencing : the Q-Element
- Micropipeline circuits

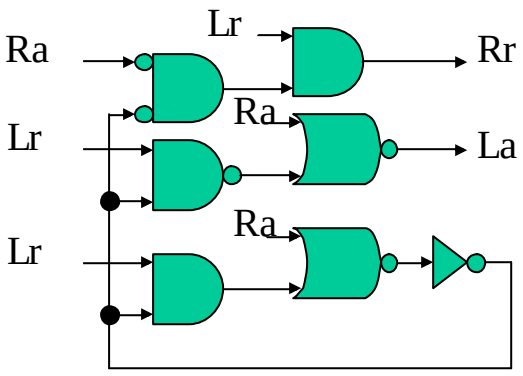
Huffman/Burst-mode asynchronous circuits

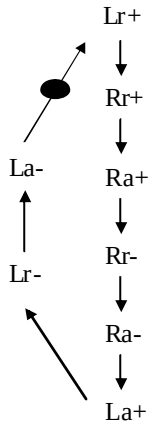
- The correctness depends on the gate/wire delays
- Based on the "bounded delay" model
- "Fundamental mode" assumption for the environment



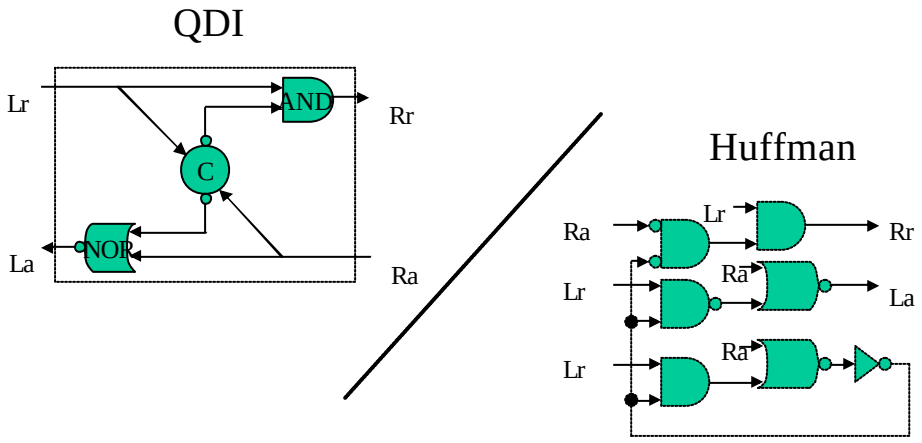
Huffman/Burst-mode asynchronous circuits

- An example : The Q-Element





Timing assumptions

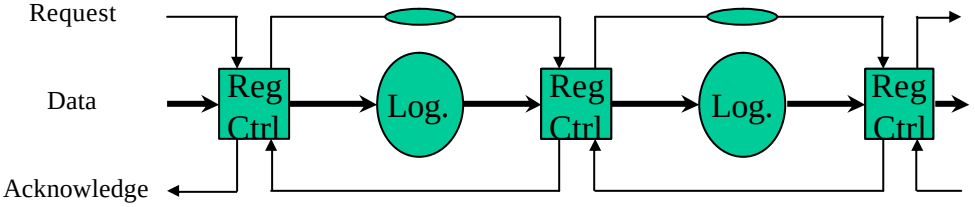


Asynchronous circuit classes

- Hazard free logic
- QDI / Speed Independent Circuits
  - Data path : a simple OR Gate
  - Sequencing : the Q-Element
- Bounded delay circuits
  - Huffman circuits / Burst mode circuits
  - Sequencing : the Q-Element
- Micropipeline circuits

Micropipeline asynchronous circuits

- Micropipeline

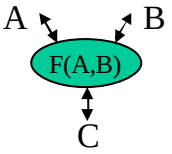


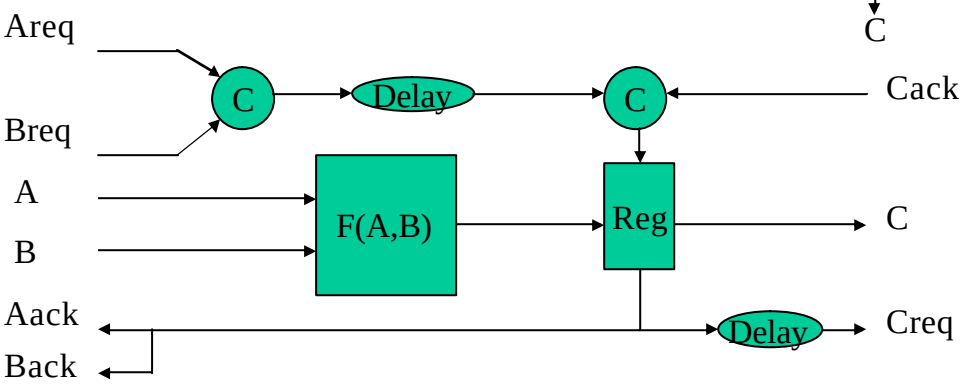
– time is discrete
→ combinatorial logic is simple
→ communication channels (handshake based)
→ worst case approach locally

Local timing assumptions

Micropipeline asynchronous circuits

- An example : function F
  hazard free control logic + standard data path





Asynchronous circuit classes

- Conclusion
  - QDI / Speed Independent circuits are the most robust with respect to delays (isochronic fork for some/all wires)
  - Huffman & Burst-Mode circuits use the bounded delay model and require fundamental mode

→ **QDI : Data-paths & Controllers**

→ **Speed Independent / Burst-Mode : Controllers**
(burst-mode controllers are difficult to compose)

→ **Micropipeline : Standard data-path + QDI/SI Controllers**

Asynchronous Circuits Design

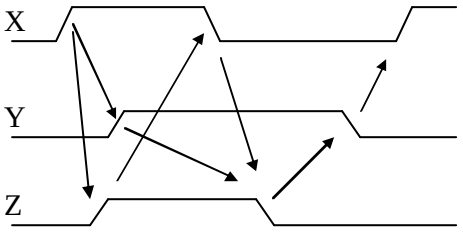
- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties
- Design experiments and products
- Conclusion and prospects

Design Methodologies and Tools

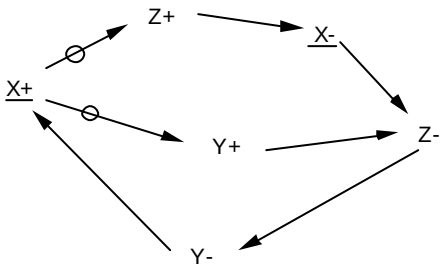
- Graph based (Petri-Nets)
  - Signal Transition Graphs
  - Burst-mode specification
- Language based (CSP based)
  - Tangram
  - CHP

Design Methodologies and Tools

- Graph based : Signal Transition Graphs

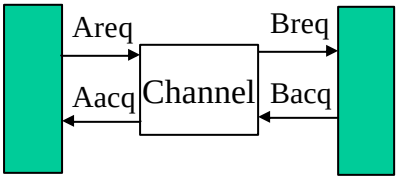


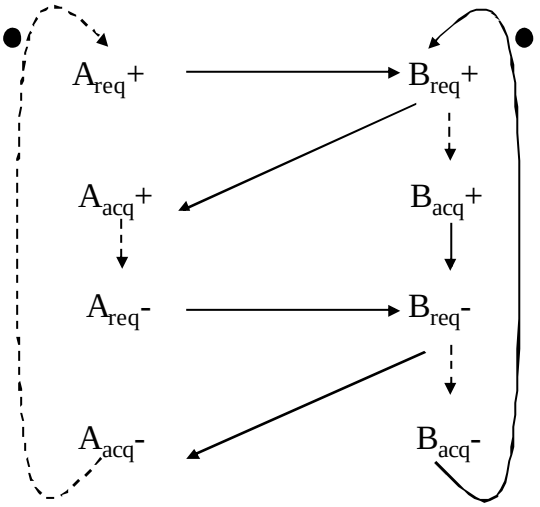
Timing diagram



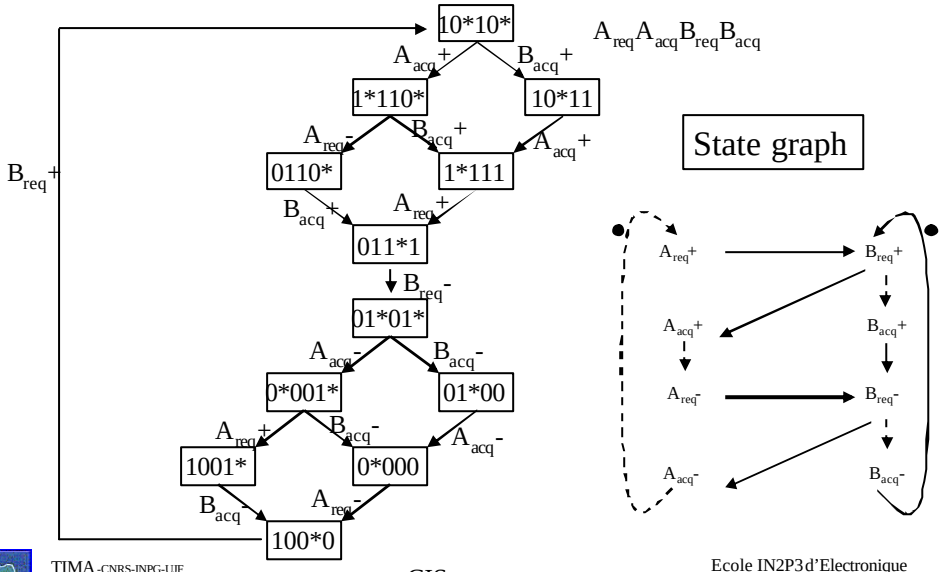
STG

Synthesis example : graph based

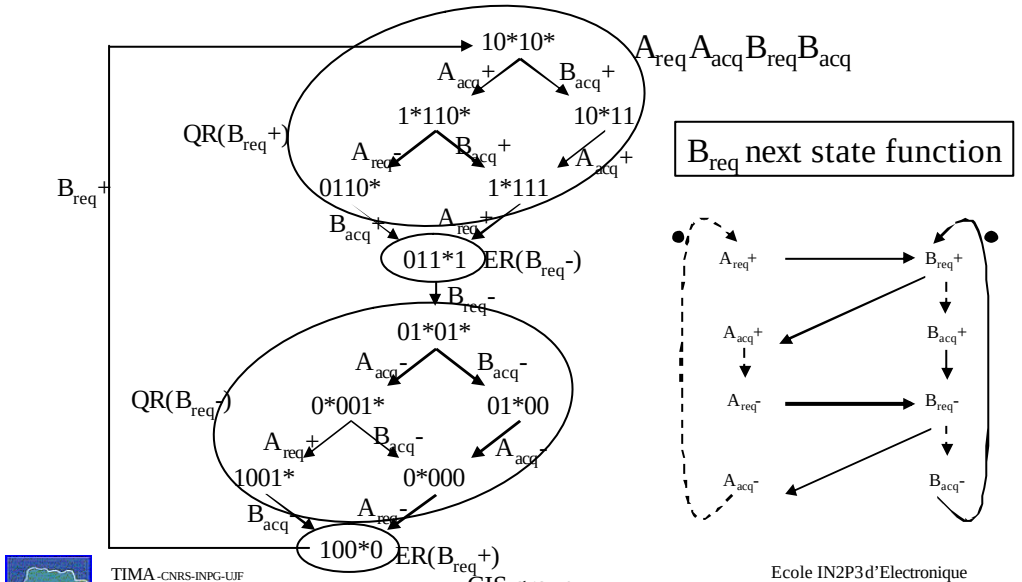




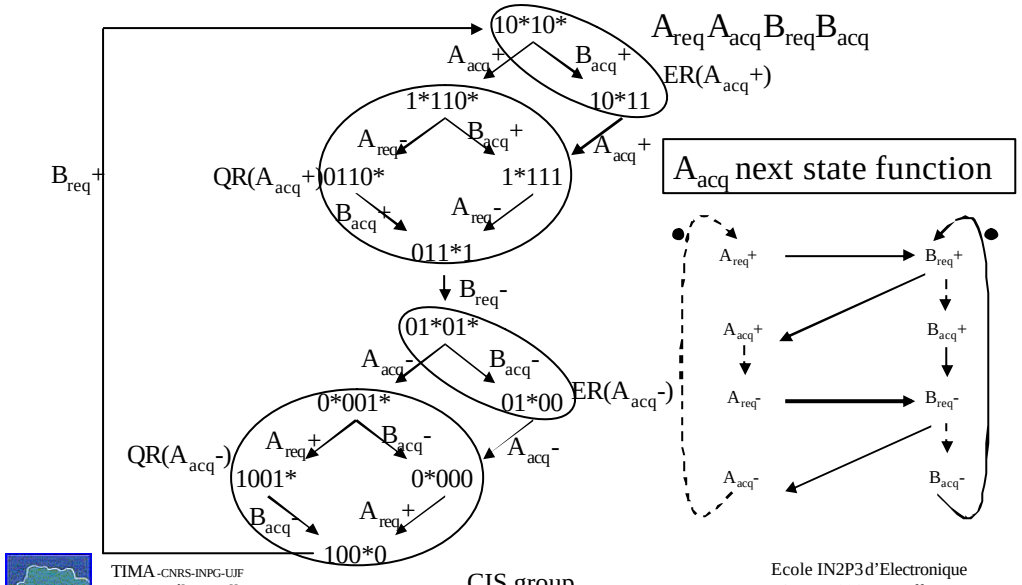
Synthesis example : graph based



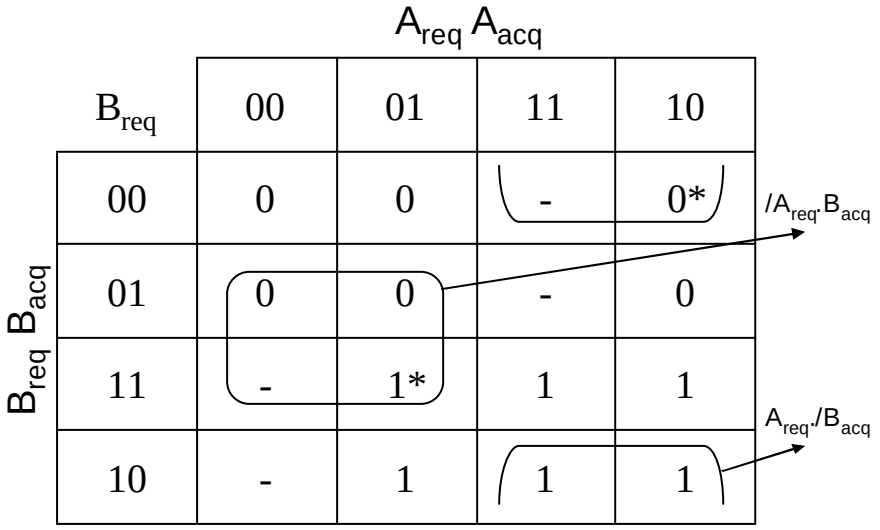
Synthesis example : graph based



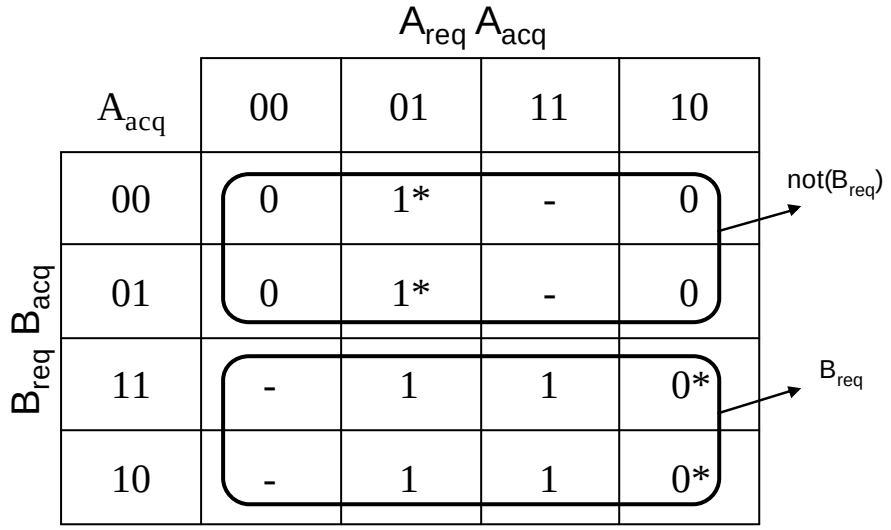
Synthesis example : graph based



Synthesis example : graph based



Synthesis example : graph based



Synthesis example : graph based

Breq

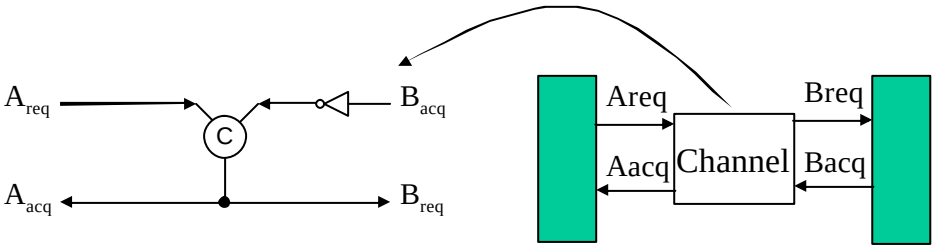
$\overline{Areq}$. Bacq => 1* => Breq ?

Areq . $\overline{Bacq}$ => 0* => Breq ?

Aacq

$\overline{Breq}$ => 1* => Aacq ?

Breq => 0* => Aacq ?



Design Methodologies and Tools

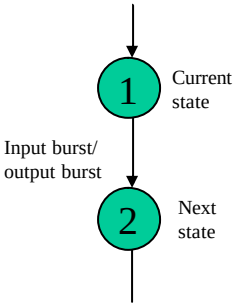
- Graph based : Signal Transition Graphs

Tools :

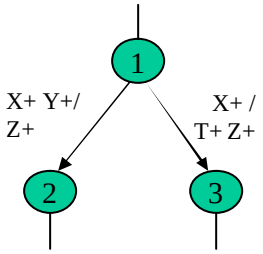
- Petrify (Barcelona)
- Atacs (Utah) (VHDL entry)
- SIS (Berkeley)
- ASSASSIN (Imec)
- ...

Design Methodologies and Tools

- Graph based : Burst-Mode specification

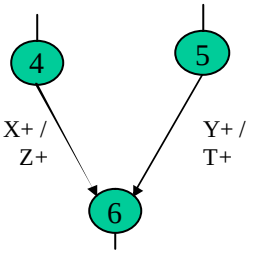


Limitations :
a) maximal set property



Ambiguous when X+

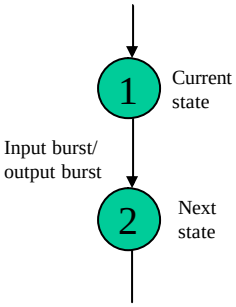
b) Unique entry point

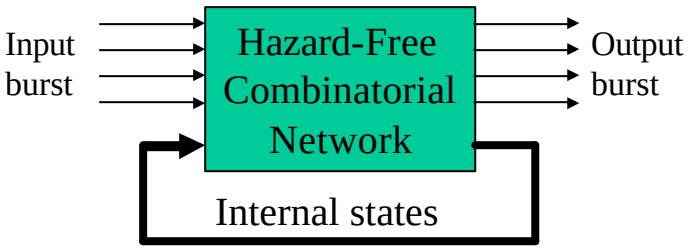


2 different input/output values when entering state 6

Design Methodologies and Tools

- Graph based : Burst-Mode specification

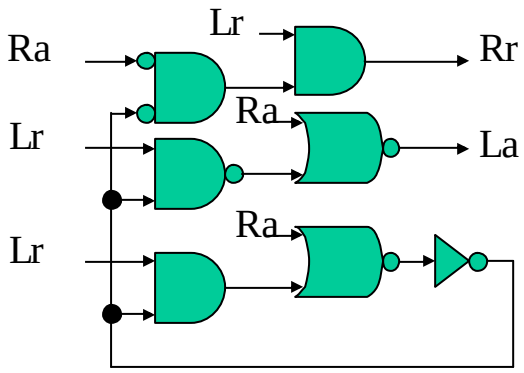


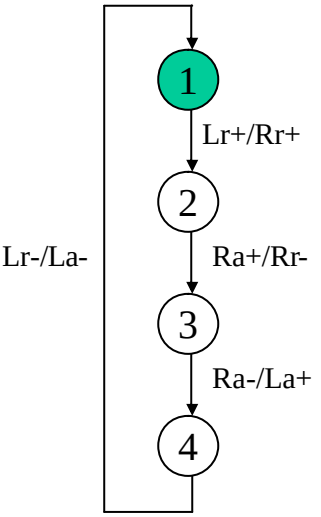


Fundamental mode required

Design Methodologies and Tools

- Graph based : Burst-Mode (Q-Element)





Design Methodologies and Tools

- Graph based : Burst-Mode Specification

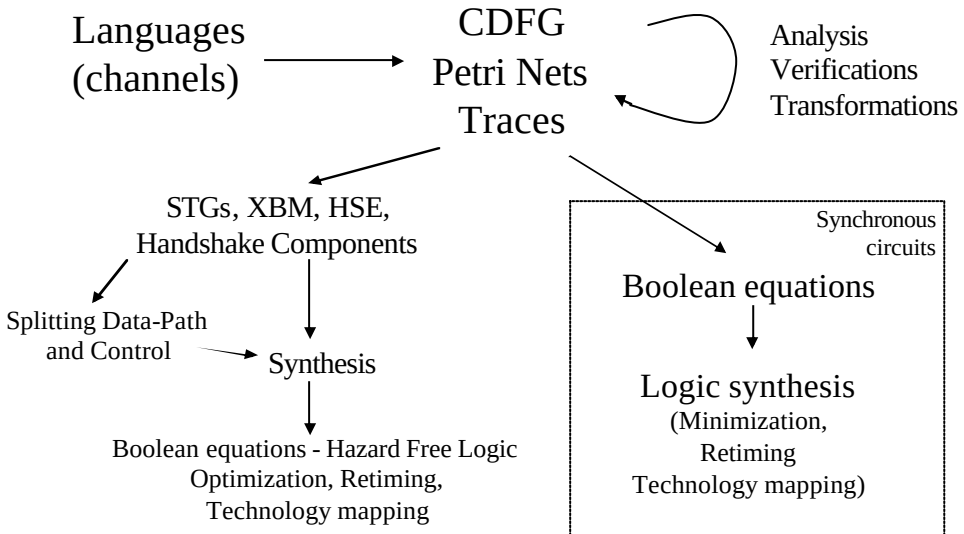
Tools :

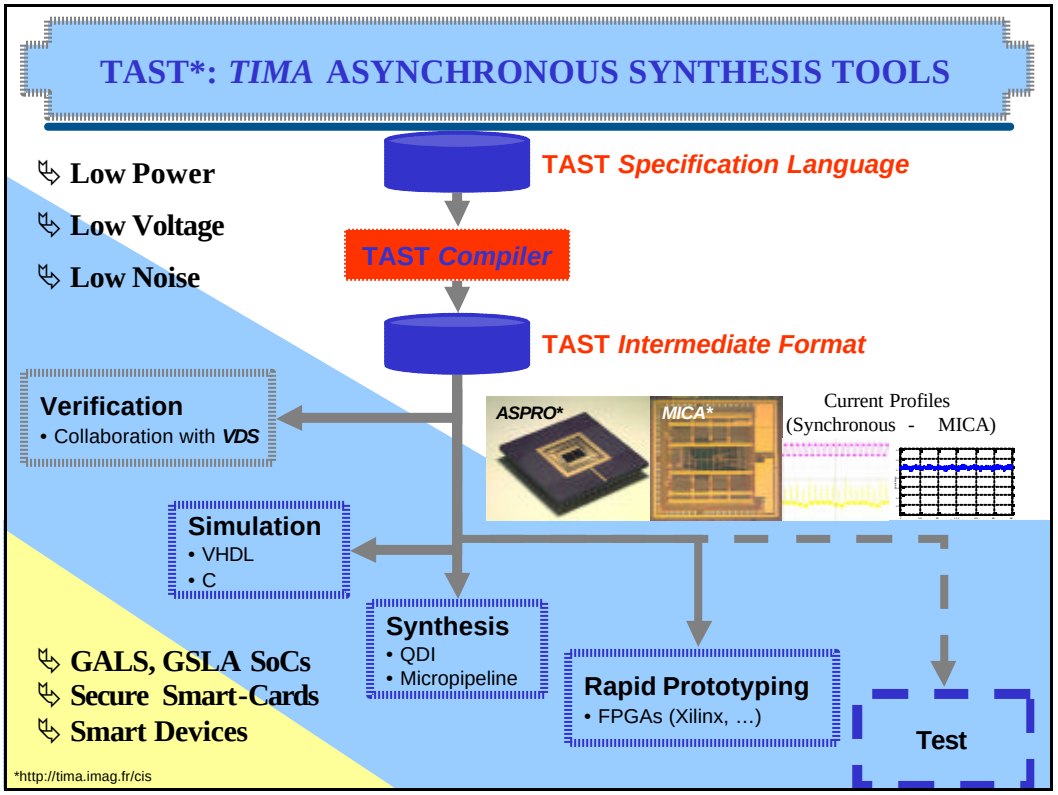
- Minimalist (Colombia University)
- 3D (USC San Diego)

Design Methodologies and Tools

- Language based
  - CSP based (Communicating Sequential Processes)
  - → CHP language (Caltech)
    - Synthesis is difficult => circuit designed can be very fast
  - → Tangram language (Philips)
    - Tool suite (Synth., Sim., Test => circuit designed are slow)
  - → Balsa language (Univ. of Manchester)
  - → TAST (Tima)
  - HDLs : modeling/synthesis are not mature yet

Design Methodologies and Tools

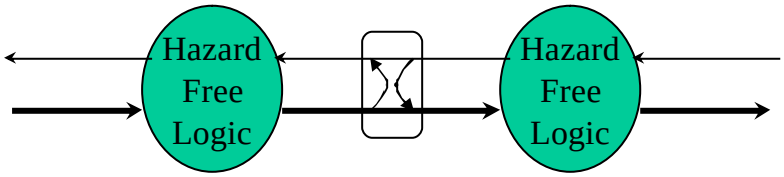




Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
- Design experiments
- Conclusion and prospects

Asynchronous Circuit Properties



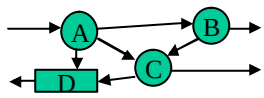
- Data flow behavior/delay insensitivity :
  => distributed control
  - minimum execution time and power consumption
  - activity and power consumption are spread in time
  - modularity and composition : reusability, IP exchange

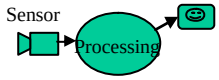
Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

Systems design requirements

- Assemble / Reuse new and existing modules
- Modules with very different architectures
- Modules activity may be very different
- Modules with different speed/power trade-offs
- Flexible on-chip communication mechanisms
- Low power
- Mix digital and Analog modules





Supervision, network...

→ Modularity and locality => reusability
- → Get rid of global constraints (like a unique global clock)
- → Avoid inheritance of constraints from block to block (like clock, noise, communication and synchronization mechanisms…)
- → Do not consume when not in use

Systems design requirements

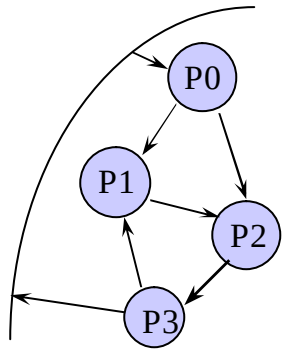
- Modularity, Locality …

→ Need to Adopt
- – the right Hardware model and
- – the right Specification model

Asynchronous Logic

and

Communicating Concurrent Processes



Asynchronous circuit properties

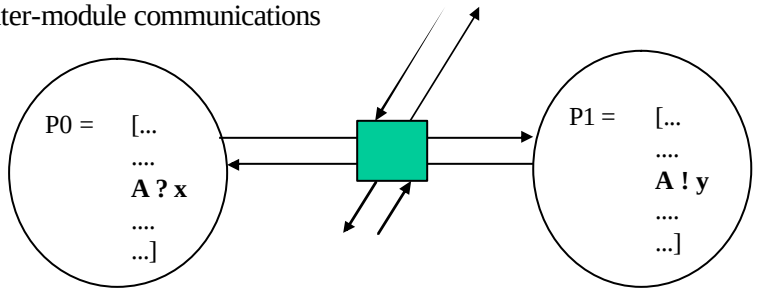
- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

Modularity and Locality

- Communication through channels (no clock)
  - ➢ Protocols are in charge of interfacing modules
    => the design of different modules is not inter-dependent
  - ➢ Design problems are solved locally inside each module
    (architecture, speed, power, back-end issues…)
  - ➢ Building a complex architecture using modules implementing handshake communication protocols is easy
  - ➢ Stand-by mode is free
  - ➢ Channels are used to trigger modules processing
  - ➢ Go from zero to maximum activity immediately, on request

Modularity and Locality

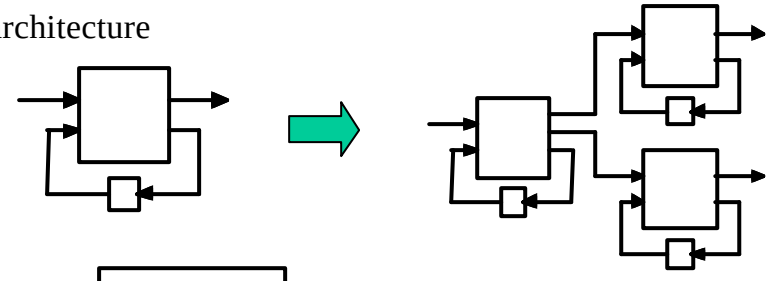
- Speed and functionality issues are solved separately
  - Within a module
  - For inter-module communications

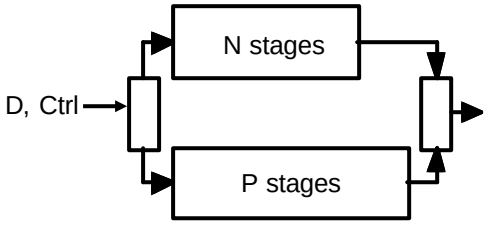


- Pipelining is preserving functional correctness in an asynchronous circuit
=> Performance optimization do not change the functionality

Modularity and Locality

- Circuit architecture





=> Locality :
<u>global state knowledge is not required !</u>

Modularity and Locality

- Thanks to asynchronous circuits
  - Provide a complete framework for jointly implementing communication channels and functions (hardware communicating processes)
  - Systems design is then easier :
    - Distributed control (protocol implementation)
    - Delay insensitive communications between modules
    - Modules are independent from each other in terms of :
      - Functionality (global state not known)
      - Speed (maximum speed)
      - Activity (power)
      - Noise (uncorrelated current consumption)
    - Scalability

Modularity
Reusability
Scalability

Asynchronous circuit properties

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

On-chip Communication Systems

- Multiple Clock Domains
  - Multiple Independent Clocks
    => Metastability may occur at clock domain boundaries

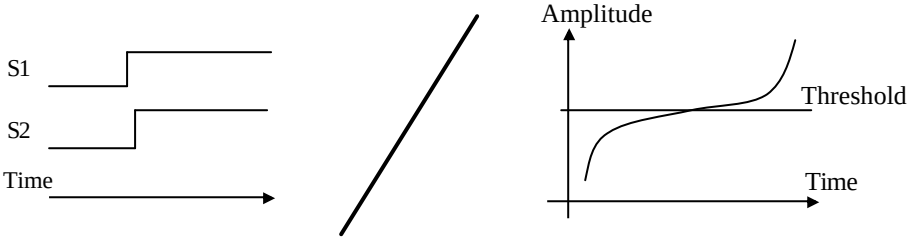
- Existing solutions
  - Control the Mean Time Between Failure (MTBF)
    - Non adaptive synchronization
    - Adaptive synchronization
  - Wait for metastability to resolve
    - Stretchable/stoppable clocks
    - Fully asynchronous (GALA)

→ Probability of error is not zero

→ Probability of error is zero

Metastability

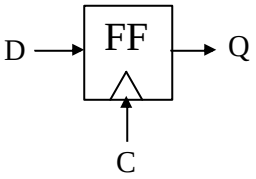
- Phenomenon and related problems
  - Any system attempting to solve one of the following problems is of limited reliability (called a synchronizer) :
    - given an input signal and a time reference, decide whether the input signal makes a transition before or after the reference,
    - given an input signal and a voltage reference decide whether the input signal voltage is higher or lower than the reference.



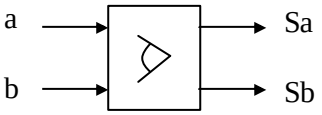
Metastability

First case :

- Asynchronous signal sampling : Synch
  - D and C are asynchronous
  - Set up and hold times may be violated
  - So what's the behavior ?



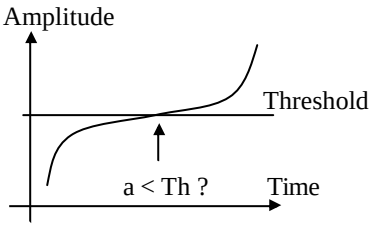
- Ordering events : ME
  - a and b are asynchronous signals
  - which signal is coming first ?
  - Sa and Sb are mutually exclusive



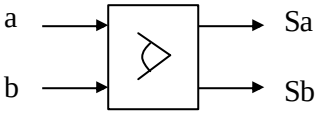
Metastability

Second case :

- Signal comparison : Synch
  - "a" is an analog signal
  - is "a" below the threshold ?
  - how long does it take to decide ?



- Ordering signals : ME
  - "a" and "b" are analog signals
  - which signal is below the other ?
  - "Sa" and "Sb" are mutually exclusive



a > b => Sa =1, Sb=0
a < b => Sa =0, Sb=1

Metastability

Where do these problems occur ?

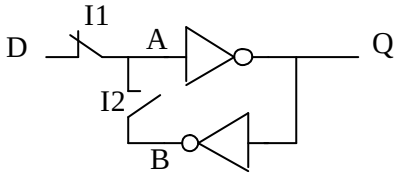
- Analog circuits
  - Comparators
  - Analog to digital converters
  - Serial links
  - CDR (clock data recovery)

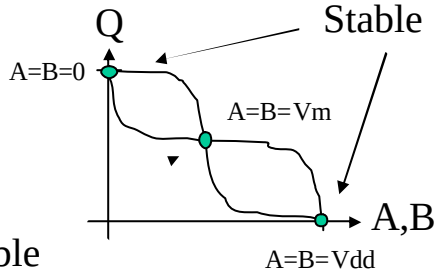
- Digital circuits
  - Processing asynchronous signals : interrupt signal
  - Communications between clock domains

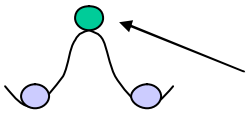
Metastability

Metastability phenomenon

- When I1 is opening and I2 is closing, we might have A=B=Vm (set up and hold times violation)
- Inner loop levels are consistent and the system may stay in this state for an infinite time in theory !
- In practice, there are perturbations which force the system to go back to one of the stable states.





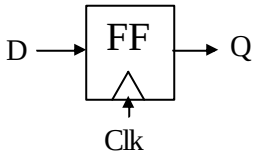


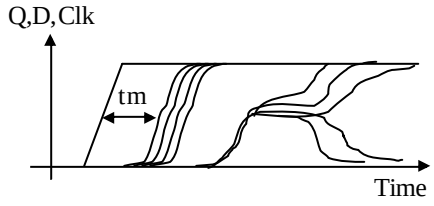
Metastable

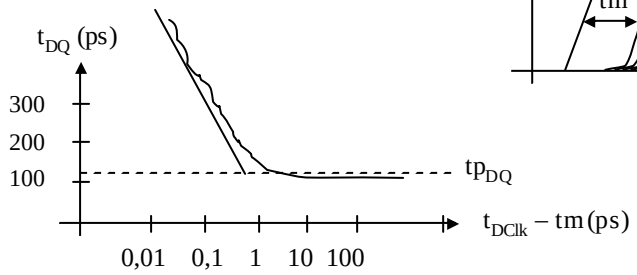
Metastability

Metastability phenomenon

- When set up and hold times are satisfied $t_{DQ} = tp_{DQ}$ bounded !
- When set up and hold times are violated t_{DQ} increases !
- In practice t_{DQ} is finite but not bounded !!







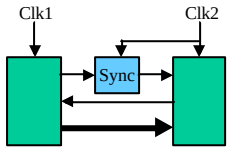
Metastability

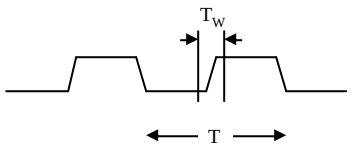
- Phenomenon model and characteristics
  - Probability that the metastable state will last longer that t' decreases exponentially with t'

$$e^{\frac{-t'}{t_r}}$$

  - Mean Time Between Failure (MTBF)
    F_c : clock frequency
    F_d : data transmission frequency

$$\frac{e^{\frac{-t'}{t_r}}}{T_W F_d F_c}$$





TW is a time window where the system may enter the metastable state (t_{DQ} increases)

Metastability

- Phenomenon model and characteristics

Some figures

.13 micron, at 200 MHz => MTBF is 10^{400} seconds.
at 1 GHz => MTBF is 10^{4} seconds (hours)

One can design a synchronizer with an MTBF longer than the life of the universe (10^{19} seconds). Is it useful ?

MTBF values depend on the applications, some hundred years is common !

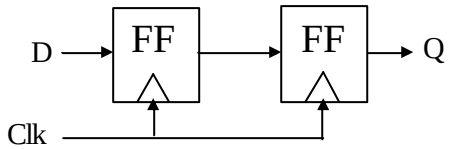
Synchronizers design issues

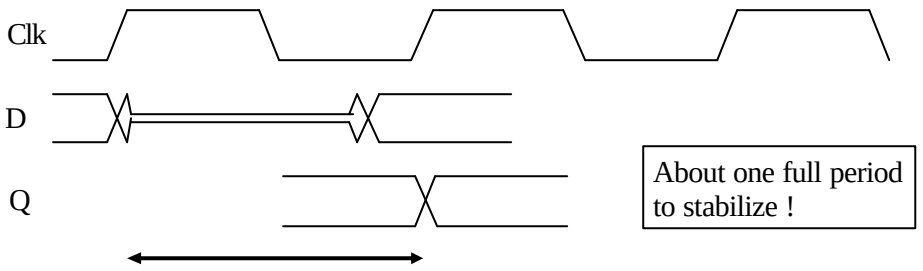
- Trends: transistor length decreases, frequencies are increasing, synchronizers design is a key know how !
- Minimize latency and failure rate

Metastability

- Phenomenon model and characteristics

A simple synchronizer

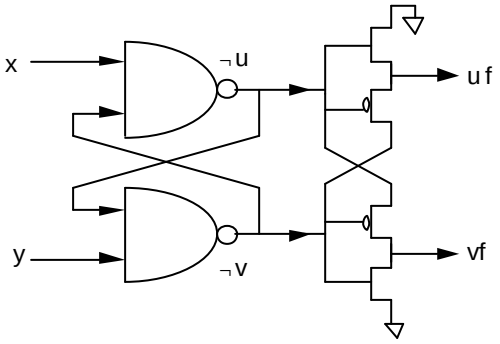




Metastability

- Phenomenon model and characteristics

A mutual exclusion circuit

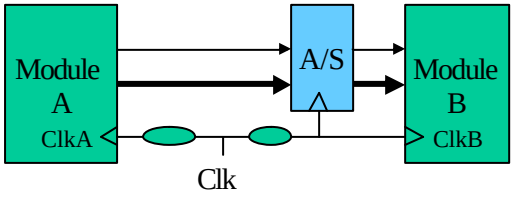


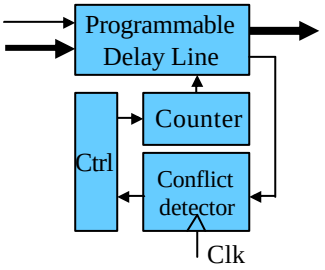
Controlling the MTBF

- How to improve the MTBF ?
  - Widening the data path
    - k times more wires
    - date rate divided by k
  - Limiting the critical control signal frequency
    - asynchronous FIFO
    - packet signaling => reduced frequency
  - Pipeline synchronization
    - adding synchronizer stages [Seizovic 94]
- Drawbacks : Latency increased
  Area and Power increased too
  → as a function of MTBF

Controlling the MTBF

- Adaptive synchronization [Ginosar 00]
  - compensate for the time-varying inter-modular clock and data phases
  - adaptation is achieved by :
    - a hardware “Adaptive Synchronizer” (A/S)
    - associated with a training session (Power-up, periodic, triggered, continuous adaptation schemes)





Controlling the MTBF

- Adaptive synchronization [Ginosar 00]
  - Estimated overhead :
    - 1% area overhead for a 1 M tr. module with 10 input busses
    - 0.1 % time overhead for a typical training session
    - less than 0.001 % power overhead
  - Advantages
    - compensate skew and drift, adaptive
    - low latency (one half cycle)
  - Drawbacks
    - need of training sessions
    - adaptive synchronizer design

On-chip Communication Systems

- Multiple Clock Domains
  - Multiple Independent Clocks
    => Metastability may occur at clock domains boundaries

- Existing solutions
  - Control the Mean Time Between Failure (MTBF)
    - Non adaptive synchronization
    - Adaptive synchronization

    → Probability of error is not zero
  - **Wait for metastability to resolve**
    - **Stretchable clocks**
    - **Fully asynchronous (GALA)**

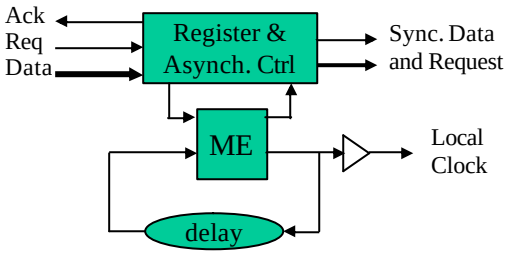
    → Probability of error is zero

Preventing Metastability

- Stretchable clocks [Chapiro 84...]
  or Pausable/Stoppable clocks [Moore 00] [Muttersbach 00]
  - based on the use of asynchronous wrappers responsible for :
    - local clock generation
    - inter-module data communication management

Asynchronous module to synchronous module interface.

Two interfaces are required for synchronous module to synchronous module communications.



Preventing Metastability

- Stretchable clocks or Pausable/Stoppable clocks
  - Ring oscillator's frequency varies with temperature and supply voltage
    → a chip cannot work at a fixed frequency
    → if required, the real time clock must be a specific module
    (or use of self calibrating clock [Moore 00])
  - Local clock buffering
  - Latency, complexity and power are all dependent on the number of channels (all processed by the same wrapper)
  - With large number of channels, conflict resolution may be long and even enter in a never-ending cyclical behavior.

Wait for Metastability to resolve

- GALA : fully asynchronous circuits
  - Modelling/Synthesis of arbitration problems [Renaudin 00]
    - from communicating processes to gates
    - requires two extra basic cells : ME and Sync
  - Safe and Fast
    - channels are independently processed
    - speed only depends of the complexity of the arbitration scheme
  - Low power
    - data driven
  - Area
    - no data buffering required
    - complexity depends on the asynchronous logic style used

Asynchronous circuit properties

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

Asynchronous On-Chip Busses design

- Fully asynchronous circuits use DI codes which have nice properties for on-chip busses design
  - One-of-N (4-phase protocols / 2-phase protocols)
    - Low power
    - Cross-talk is reduced
    - Area is about the same
    - Speed is increased
      - Electrical buffering and pipelining at the same time
      - Safe, fast and low-power arbiters can be designed

Asynchronous On-Chip Busses design

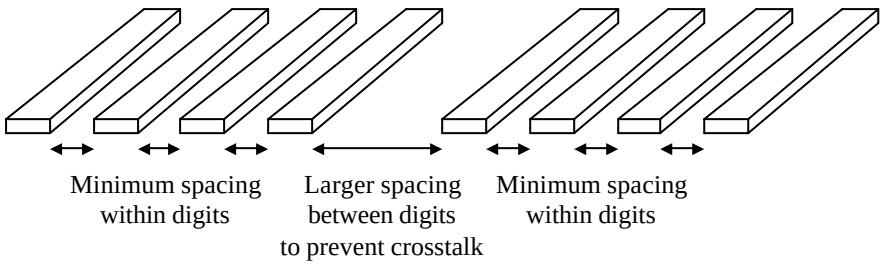
- 1-of-4 DI code energy efficiency

	Wires/bit	Transitions/bit
single rail	1	1/2 on average
dual-rail 4-phase	2	2
dual-rail 2-phase	2	1
1-of-4 4-phase	2	1
1-of-4 2-phase	2	1/2

Same Wire/bit than dual-rail code but more energy efficient

Asynchronous On-Chip Busses design

- DI codes for cross-talk reduction



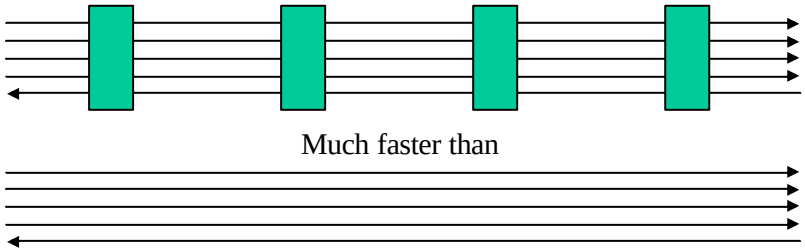
We know where cross-talk is likely to happen

Area overhead is small even if the number of wires is doubled

Acknowledge signals may be used as spacers

Asynchronous On-Chip Busses design

- DI code for long wires routing
  - Repeaters perform electrical buffering and pipelining at the same time



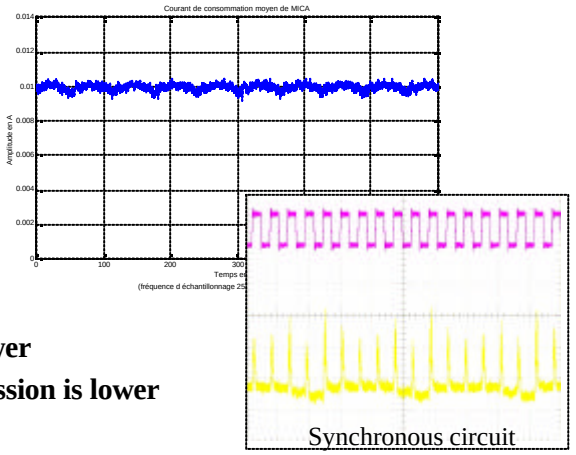
  - Safe, fast and low-power arbiters can be designed

Asynchronous circuit properties

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise and current shaping
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

Noise

Measurements performed with "MICA" an asynchronous QDI 8-bit microcontroller
Current profile (2000)



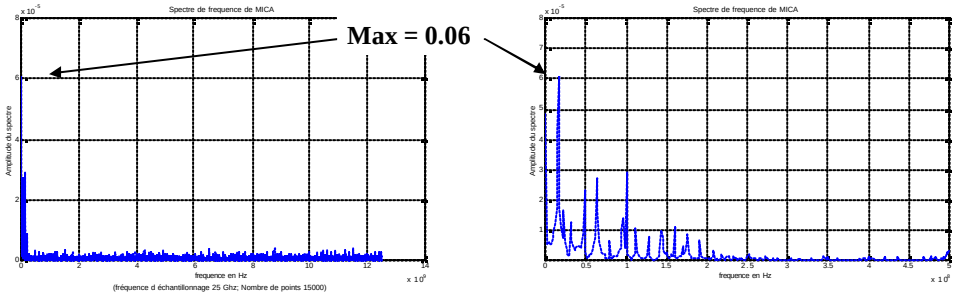
- **10 mA average current**
- **0.6 mA amplitude variations**

Mean power consumption is lower

Smaller current peaks , EM emission is lower

Noise

Measurements performed with "MICA" an asynchronous QDI 8-bit microcontroller
Current spectrum (2000)

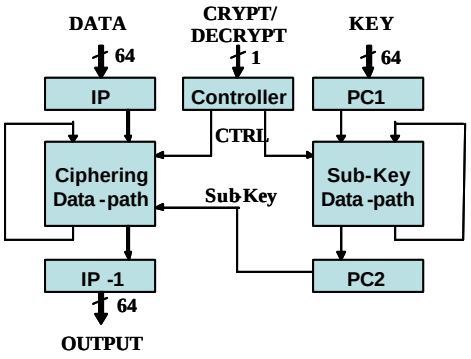


0 to 12.5 GHz

0 to 0.5 GHz

Noise

- Data Encryption Standard (DES) crypto-processors



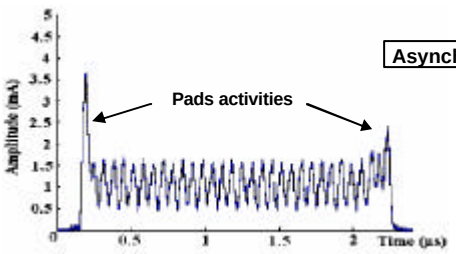
Asynchronous DES Chip architecture

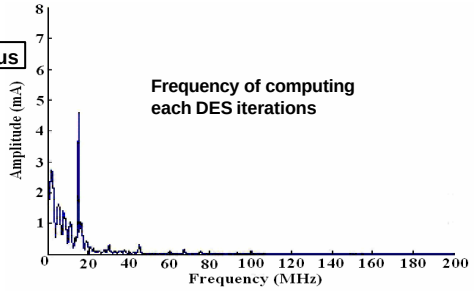
Circuits' Layout

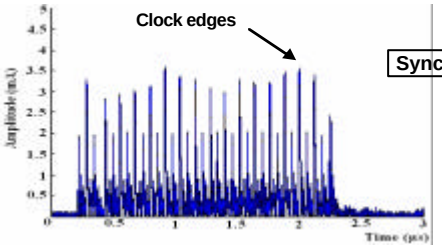
Synchronous **Asynchronous**

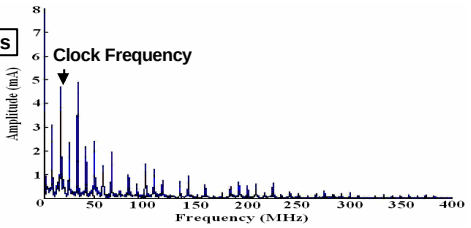
CMOS 0,18 µm (HCMOS8) from STMicroelectronics
6-LM, Power Supply 1.8 V

Results : DES Chip







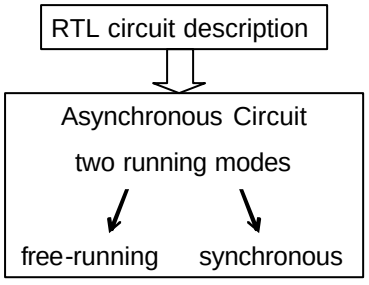


Asynchronous circuit properties

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise and current shaping
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

Current shaping

- Power aware scheduling (thèse de Dhanistha Panyasak)



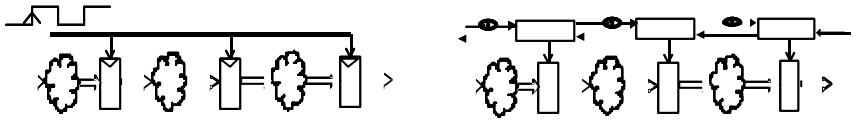
Le circuit est cycle compatible

et faible bruit, faible rayonnement

Contrôler le niveau de bruit dans la bande utilisée par le front-end RF

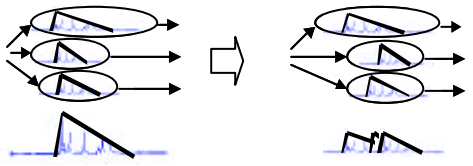
Power Aware Scheduling

- By relaxing synchronization between blocks :



…use of low EMI advantage of Asynchronous Circuits !

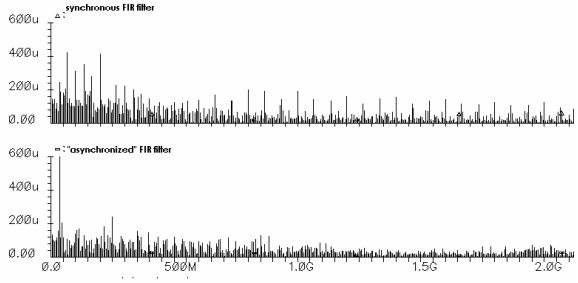
- By scheduling events :



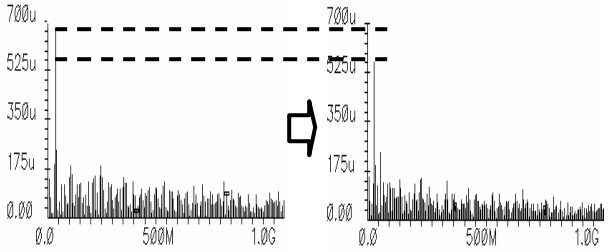
… spread spectrum of handshaking !

Power Aware Scheduling

RTL code
to
Asynchronous Circuits

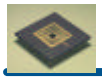


Applying
scheduling
(FDS)



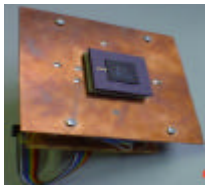
Asynchronous circuit properties

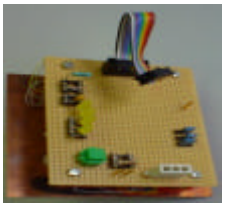
- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

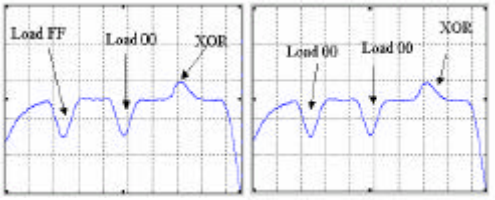


Security

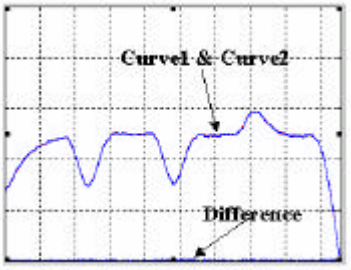
- Processor MICA





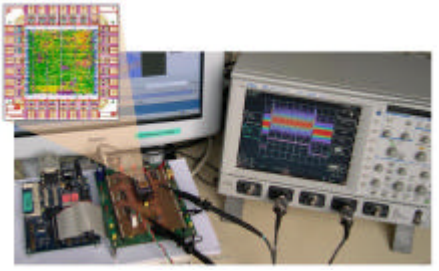


- Number of points : 100000
- 10000 computations



Tamper-resistant hardware

- Hardware cryptanalysis (measurements, post-processings)
- Counter measures design using ASL
  - PA
  - DFA
- Prototyping ASL
  - Asic : DES, AES
  - FPGAs
- Methodologies and CAD tools for improving PA and DFA chip resistance using ASL



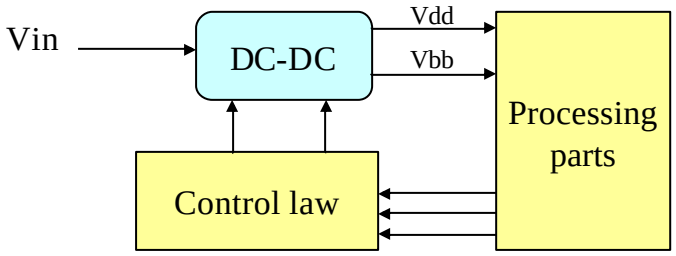
Asynchronous circuit properties

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
    - Noise
    - Security
    - Automatic performance regulation
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

Self adapting performance

® **Minimum energy computation**

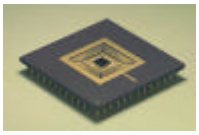
Processing and data have irregular natures (exploiting the dynamic)
A breakdown with respect to the synchronous approach => algorithms must be different (just in time processing)

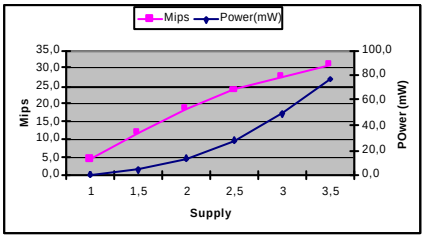


Cf.
Microprocessor speed control using Vdd

8-bit CISC Asynchronous Microcontroller : Vdd range

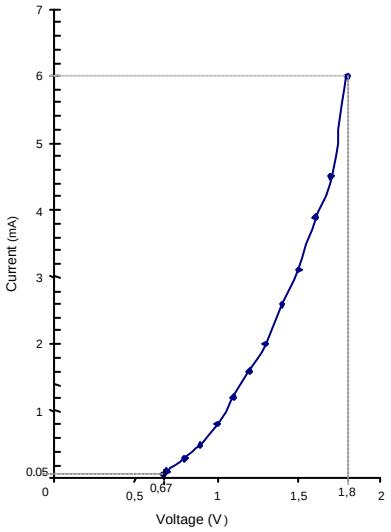
- 1-of-4 DI codes for arith. and reg.
- 1-of-n DI codes for the control
- Complexity
  - 145 000 transistors, 0.25 µm STM
  - 1 M transistors with memories
  - 13 mm² with pads (prototype)
  - PGA120 package for the prototype
- Test
  - BIST (approx. 300 instr)
  - functional al 1er silicium between 3v et 0.65 v
- 24 Mips / 28 mW @ 2.5V
- 4,3 Mips / 0.8 mW @ 1V

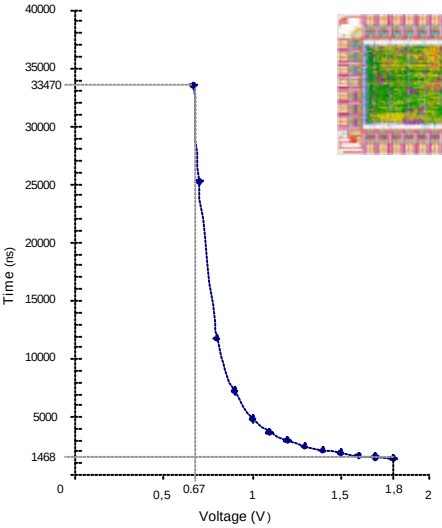




Supply(V)	Mips	Core Current (mA)	Power(mW)	Mips/Watt
1	4,3	0,8	0,8	5503,6
1,5	11,9	3,1	4,7	2560,2
2	18,6	6,7	13,3	1398,0
2,5	23,8	11,2	28,0	850,3
3	27,8	16,3	48,9	568,1
3,5	31,3	22,0	77,0	405,8

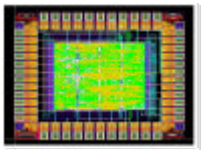
DES chip (HCMOS8) : Vdd range

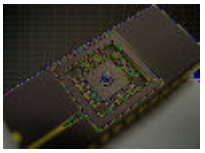


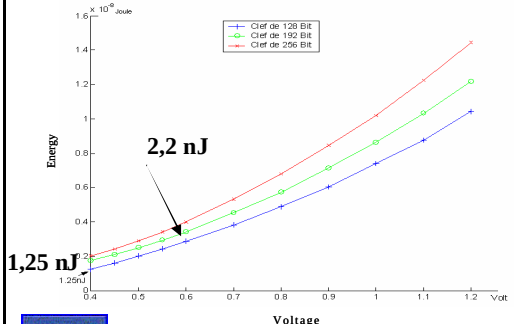


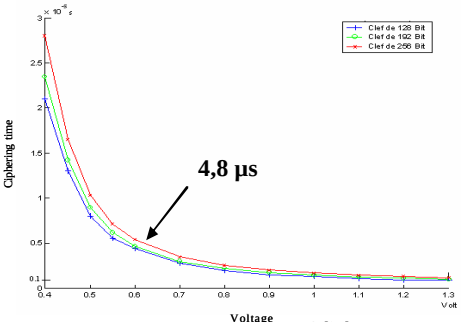
Asynchronous AES crypto-processor

	CMOS 0,13 µm (HCMOS9) from STMicroelectronics 6-LM, Power Supply 1.2 volt				
Key length	Area (core)	Area with pads	Ciphering Time (ns)	Energy (nJ)	Throughput
128-bit	0.49 (mm²)	1.69 (mm²)	910	10	141 Mbits/s
192-bit			1.100	12	116 Mbits/s
256-bit			1.440	14,2	89 Mbits/s





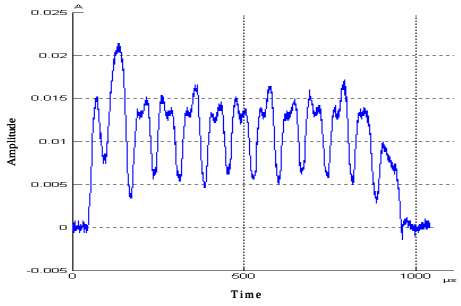




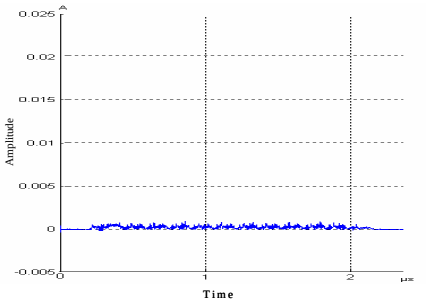
Asynchronous AES crypto-processor

- Current profiles

Powered at 1.2 volt



Powered at 0.4 volt

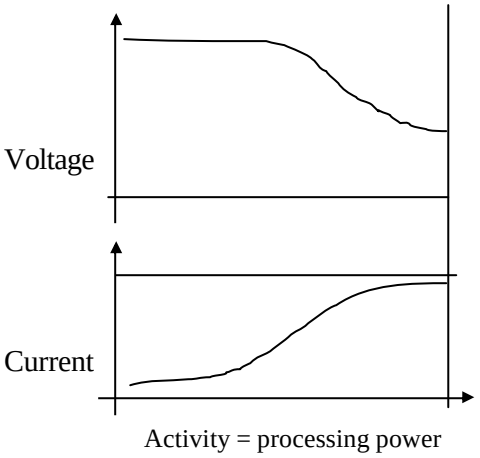


Limited power budget

- Power supply controlled systems
- Processing power is limited by the power budget available

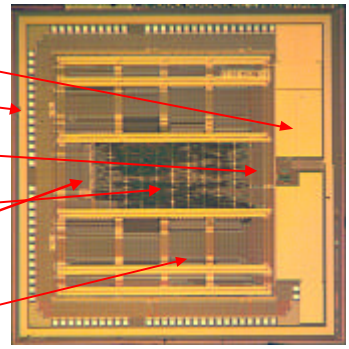
→ Maximum performance delivered with the available power received

Applications :
remotely powered systems (RFIDs)



Contactless Smart-Card Chip

- "SoC" for Contactless Smart-Card
  - Power reception system (on-chip coil)
  - ISO 14443-B std compliant
  - 8-bit CISC Asynchronous Microcontroller designed with standard cells (Mica)
  - Rom
  - Rams



Collaboration with France Telecom/R&D
Cmos 0.25 µm STMicroelectronics
[IEEE-JSSC July 2001]

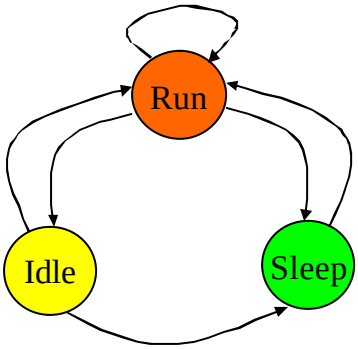
Asynchronous circuit properties

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
  - Systems design requirements
  - Modularity and locality
  - On-chip communication systems
  - Asynchronous on-chip busses design
  - Current consumption profile
  - Dynamic voltage scaling
- Design experiments
- Conclusion et prospects

Voltage Scaling : sync. vs async. !

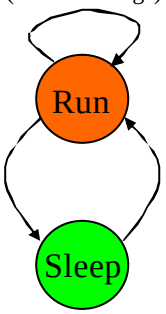
Synchronous

Vdd, Fclk



Clock-less

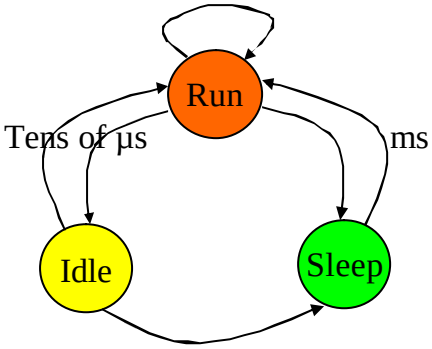
Vdd
(extended range)



Voltage Scaling Overheads !

Synchronous

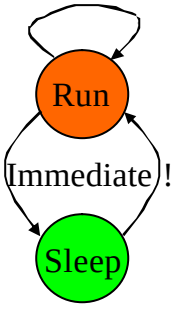
Tens to hundreds µs



Tens of µs

Clock-less

DC-DC faster than PLL

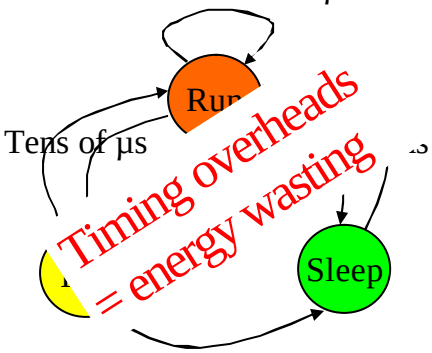


Circuit keeps running
at maximum speed !

Voltage Scaling Overheads !

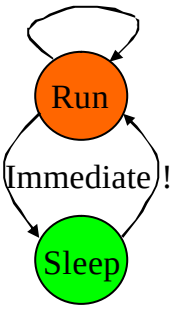
Synchronous

Tens to hundreds µs



Clock-less

DC-DC faster than PLL

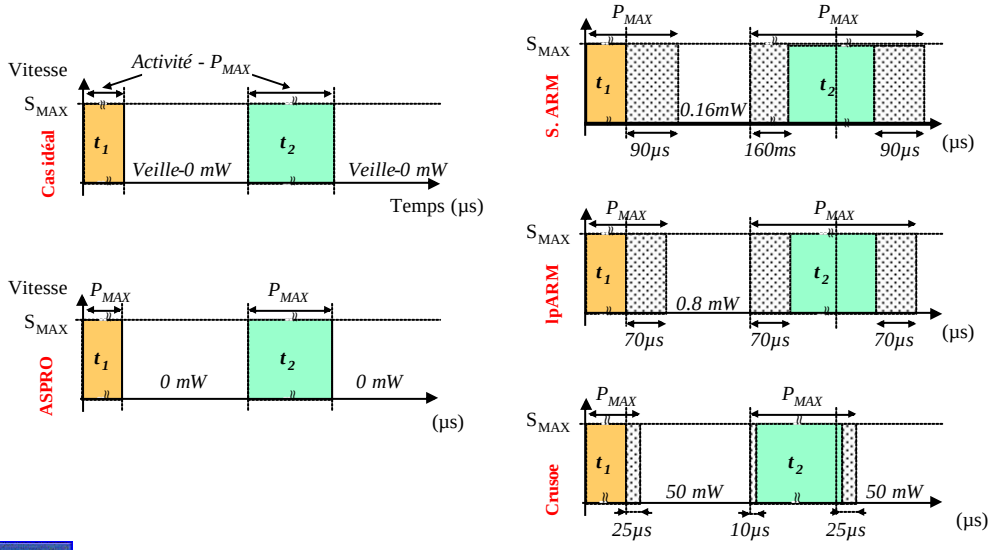


Circuit keeps running
at maximum speed !

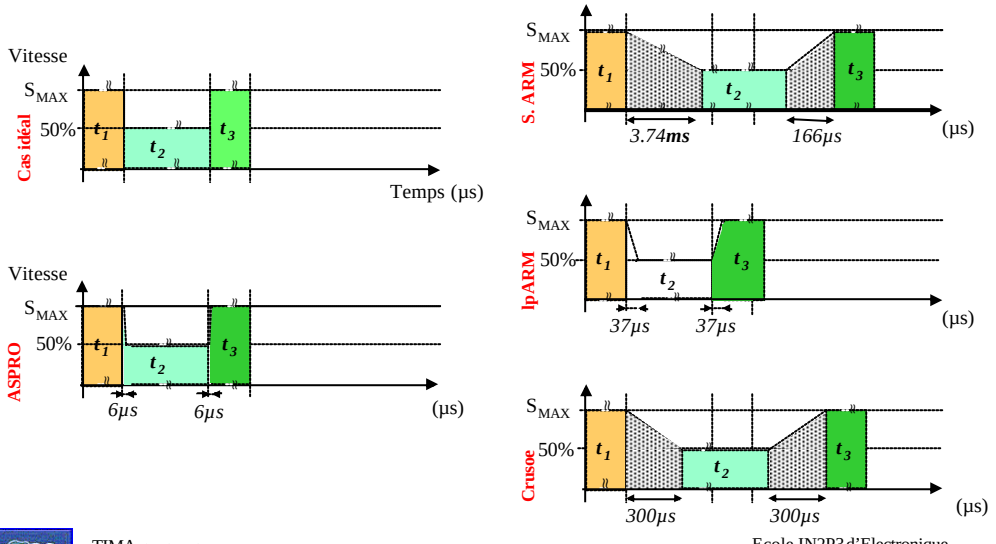
Asynchronous vs Synchronous : synchronous processor performance

- StrongARM-1100
  - 59MHz - 0.79V et 251MHz - 1.65V
  - Un changement de fréquence prend 140µs
  - Un changement de tension prend 40µs
  - Mise en veille : 90µs - Réveil : 160 ms
- lpARM (ARM8 Berkeley)
  - 6Mips, 5MHz - 1.2V et 85Mips, 80MHz - 3.8V
  - Un changement de fréquence de 5MHz à 80MHz prend environ 70µs
  - Veille (idle) : 0.8mW
- Transmeta Crusoe
  - 300MHz - 1.2V à 600 MHZ - 1.6V
  - Un changement de fréquence de f_{min} à f_{max} prend 300 µs
  - pas de 33 MHz, 25 mV

Asynchronous vs Synchronous : standby mode

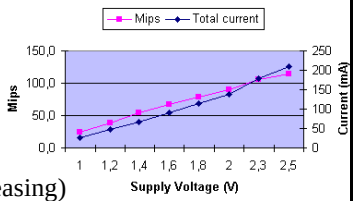


Asynchronous vs Synchronous : speed adaptation



Dynamic Voltage Scaling/Scheduling

- Benefits of an asynchronous processor :
  - No software to manage the processor or the peripherals activity (run, idle, sleep)
  - Consumption in sleep mode = consumption in idle mode (ratio higher than 100 for the synchronous processors)
  - Wake up/stop at very high frequency for the processor and the peripherals (thousands of cycles for a synchronous processor)
  - Speed and energy controlled by the voltage only (switching time much smaller : factor of 2 to 5)
  - Large voltage range (more and more critical with the supply voltage decreasing)



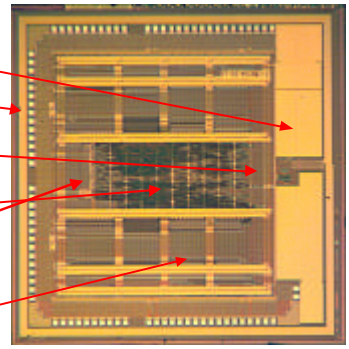
ASPRO

Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
- Design experiments
  - Contact-less Smart Card
  - ASPRO RISC microprocessor
  - Towards fully asynchronous systems
- Conclusion and prospects

Contactless Smart-Card Chip

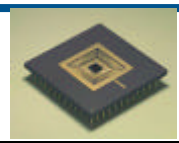
- "SoC" for Contactless Smart-Card
  - Power reception system (on-chip coil)
  - ISO 14443-B std compliant
  - 8-bit CISC Asynchronous Microcontroller designed with standard cells (Mica)
  - Rom
  - Rams

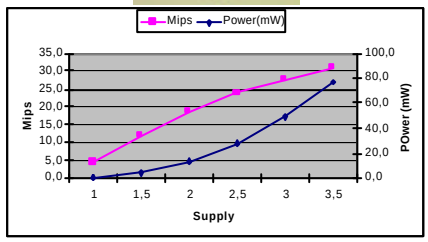


Collaboration with France Telecom/R&D
Cmos 0.25 µm STMicroelectronics
[IEEE-JSSC July 2001]

Mica : an 8-bit CISC QDI Asynchronous µC

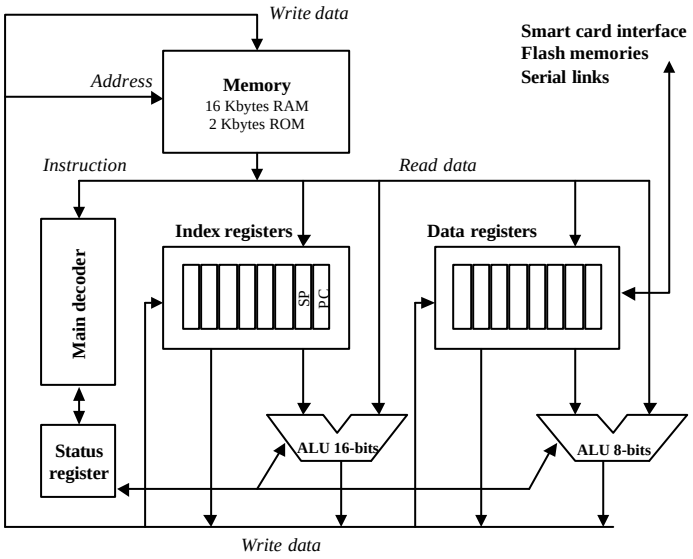
- 1-of-4 DI codes for arith. and reg.
- 1-of-n DI codes for the control
- Complexity
  - 145 000 transistors (0.25 µm)
  - 1 M transistors with memories
  - 13 mm² with pads (prototype)
  - PGA120 package for the prototype
- Test
  - BIST (approx. 300 instr)
  - functional at 1er silicon between 3v et 0.65 v
- 24 Mips / 28 mW @ 2.5V
- 4,3 Mips / 800 µW @ 1V





Supply(V)	Mips	Core Current (mA)	Power(mW)	Mips/Watt
1	4,3	0,8	0,8	5503,6
1,5	11,9	3,1	4,7	2560,2
2	18,6	6,7	13,3	1398,0
2,5	23,8	11,2	28,0	850,3
3	27,8	16,3	48,9	568,1
3,5	31,3	22,0	77,0	405,8

MICA : Architecture

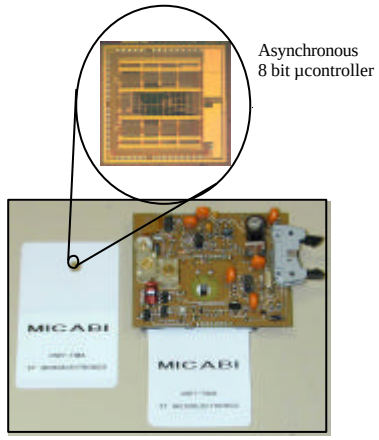


MICA : Instruction Set

Arithmetic/logic	**Register** **Add, Addc, Sub, Subb, Inc, Dec, Neg** **And, Or, Xor, Not,** **Cb*n* (clear bit *n*), Sb*n* (set bit *n*), Tb*n* (test bit *n*)** **Rol, Ror, Rcl, Rcr (rotate without and with carry)** **Shl, Shr, Shrs (shift left, shift right unsigned and signed)** **Index** **Addx, Subx**
Load/Store	**Register** **Ld, Ldp, Stp (load, store peripheral), St** **Cp (copy from memory to memory is available)** **Pl (push & load)** **Psh, Pshsr (push, push status register) , Pop** **Index** **Pshx, Popx**
Control flow	**Rti, Rts, Jmp, Jsr** **Bcc, Bsrcc (cc = a,eq,ne,cc,cs,l,le,lt,lte,g,ge,gt,gte,vc,vs)**
Misc	**Nop, Wfi, Eint, Dint (enable and disable interrupt)**

Contactless Smart-Card Chip

- Asynchronous Logic relaxed Design constraints
  - Not sensitive to supply voltage variations
    - →Power reception system (capacitances area, voltage regulation)
  - Lower current peaks
    - → The Micro-controller can be running during the communications without disturbing the load modulation.
  - Maximum processing power delivered according to the power received



Collaboration with France Telecom/R&D

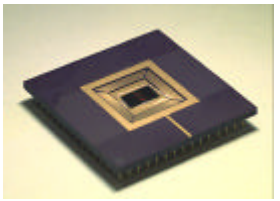
Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
- Design experiments
  - Contact-less Smart Card
  - ASPRO RISC microprocessor
  - Towards fully asynchronous systems
- Conclusion and prospects

ASPRO : a RISC Microprocessor

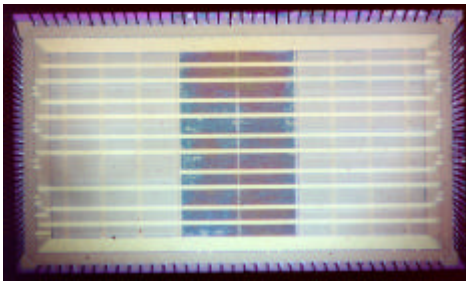
- A QDI Asynchronous 16 Bit RISC Microprocessor

=> Performance
=> Standard-Cells
=> Ease of design (4 m.y)
- Architecture
(out of order completion)
- System Board



ASPRO

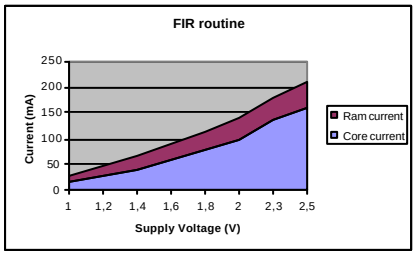
- QDI asynchronous logic
- Standard Cells
- 500 KTr for the core
- 6.3 MTr with memories
- Total area is 42 mm^2
- CMOS 0.25µm 6 metal layers STMicroelectronics

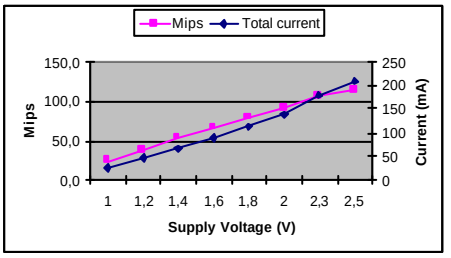


- Use of standard tools except
  - CHP2VHDL translator
  - Synthesis

ASPRO

- Functional at first silicon : 0.65V and 3.0V
- 140 MIPS (max)
- ASPRO includes DSP capabilities (MACC unit, brv...),
  RIF routine runs at 115 MIPS, 500 mW (including memories)
- Serial links (2 phase) : 50Mbit/s





ASPRO

- **Processor's main features**

Memories
- **program : 16 kwords on chip**
  48 kwords off chip
- **data : 64 kbytes on chip**

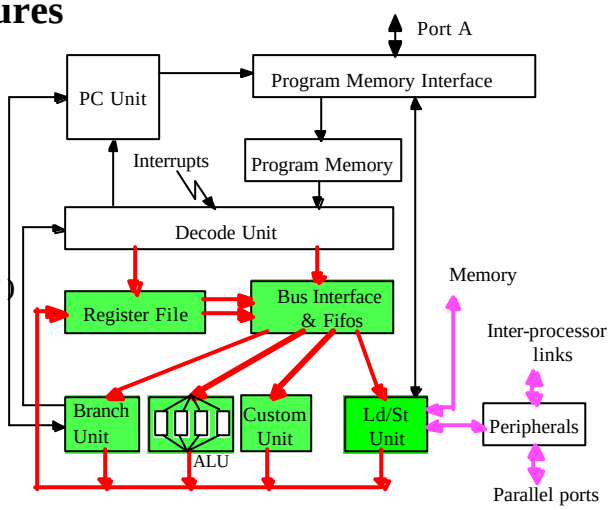
16 general purpose registers

On chip peripherals
- **2 parallel ports**
- **4 bidirectional serial links (2F)**

Custom units
- **added to the peripheral area**
- **embedded in the data path**

Three supply voltages

Interrupt mechanism



ASPRO

- **ASPRO's instruction set**

Alu instructions : std arithmetic, logic and shift/rotate
- **min/max, bit reverse, slt/sltu (no status register)**

Load/Store instructions : byte/word load and store
- **basic addressing mode is indirect with displacement**
- **immediate load is provided (16 bit values)**
- **load relative address (relocatable code)**
- **program memory load/store through a dedicated register (boot)**

Program flow instructions
- **conditional relative branch (eq,ne,lt,ltu)** **=> delayed or not**
- **djmp, dbsr, djsr** **=> delayed**

Custom instructions : 64 slots
- **mpy, macc (integer, fixed point, rounding and saturation modes)**
  (16 x 16) + 40 => 40 bit

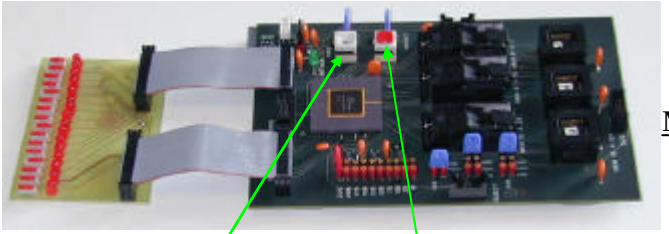
ASPRO : a RISC Microprocessor

- ASPRO
- Flash memories
- Reset/Interrut logic
- Peripherals : 2ϕ Serial Links & Parallel Ports

=> A multi-processor system

Daughter board

Switches
&
LEDs



Mother board

Interrupt Reset

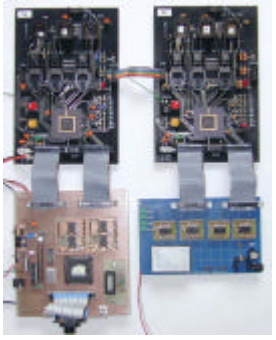
Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
- Design experiments
  - Contact-less Smart Card
  - ASPRO RISC microprocessor
  - Towards fully asynchronous systems
- Conclusion and prospects

Towards Fully Asynchronous Smart Devices

- Camera
- Image processing
- RF communication

→ Integration on a single chip
  - Modular
  - Event driven
  - Low power
  - Low noise



Asynchronous Circuits Design

- Asynchronous circuit design principles
- Basic structures and asynchronous circuit classes
- Design methodologies and tools
- Asynchronous circuits properties and potentials
- Design experiments
- Conclusion and prospects

Conclusions et prospects

Delay insensitivity

- ? circuits are insensitive to IR drop phenomenon
- ? circuits are insensitive to PVT variations
- ? characterization efforts can strongly be reduced

This is a serious solution to « easily » and « rapidly » design IPs using advanced CMOS processes, 65nm, 42 nm …

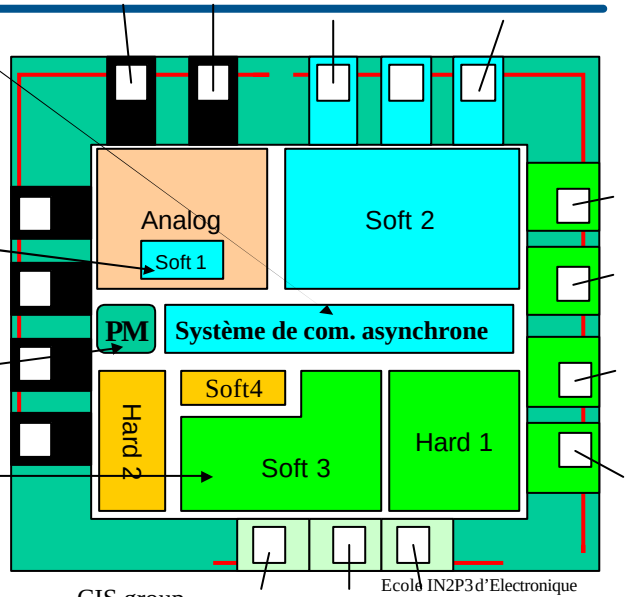
Asynchronous circuits in SoCs

Asynchronous communication systems
- **flexible (clock, power)**
- **performance/cons.**
- **transport/synchronization**

Logic functions embedded in analog blocks
- **low noise/power.**
- **no clock**

Power Management
- **Voltage scaling (Vdd, Vt)**

Soft asynchronous blocks
- **power**
- **security**
- **noise**



Conclusion and Prospects

- High quality/complexity demonstrators exist
- Commercial products available
- Some new are coming (smart card)
- Companies (Handshake solutions, Theseus Logic, Fulcrum, Silistix, Achronix)

→ Strong need of efficient CAD Tools !

→ Asynchronous circuits testing is still a major issue !

Links

- TIMA CIS group
  http://tima.imag.fr/cis/
- Asynchronous bibliography :
  http://www.win.tue.nl/cs/pa/wsinap/async.html
  http://www.cs.man.ac.uk/amulet
- Tools :
  http://www.lsi.upc.es/~jordic/petrify
  http://www.cs.columbia.edu/async/minimalist/minimalist_homepage.html
  http://www.cs.man.ac.uk/amulet/projects/balsa
  http://www.async.elen.utah.edu/

Links

- Commercial circuits :
  http://www.sharpsdi.com
  http://www-us.semiconductors.com/pip/PCA5007H
  http://www-us.semiconductors.com/pip/PCA5010H
- European "**A**synchronous **Ci**rcuit **D**esign" Working Group :
  http://www.scism.sbu.ac.uk/ccsv/ACID-WG