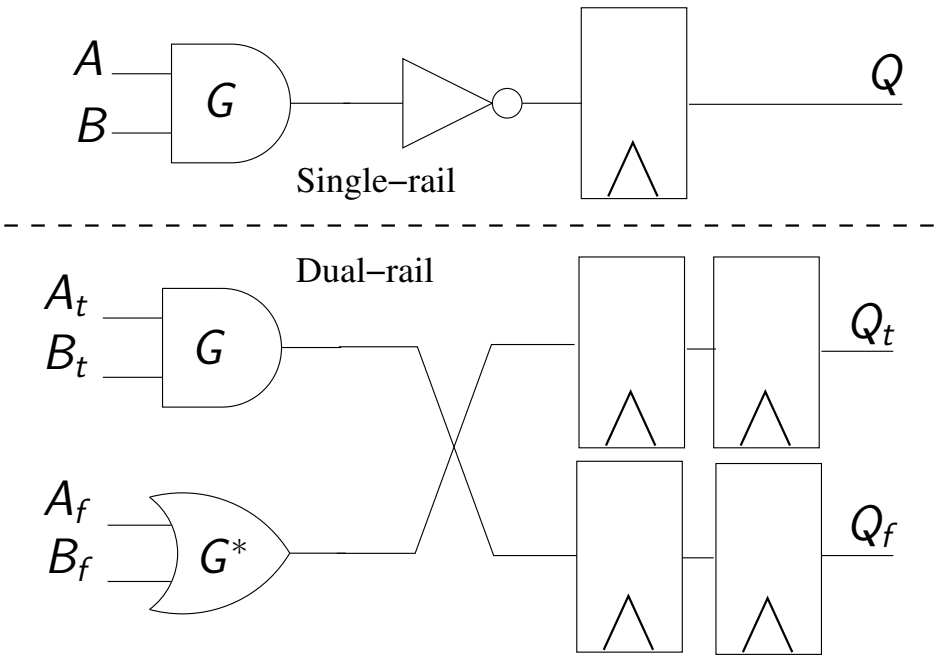
BCDL Logic design with the best Trade-off Complexity/Robustness

CRYPTARCHI 2011

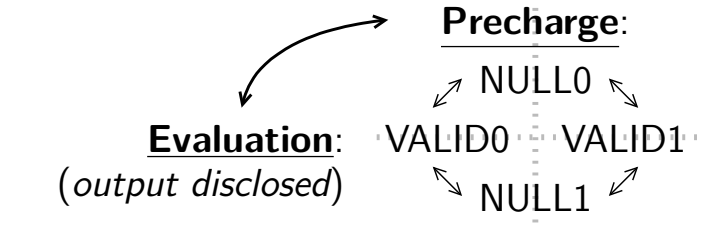
Telecom ParisTech, Secure-IC, Thursday, June 16, 2011

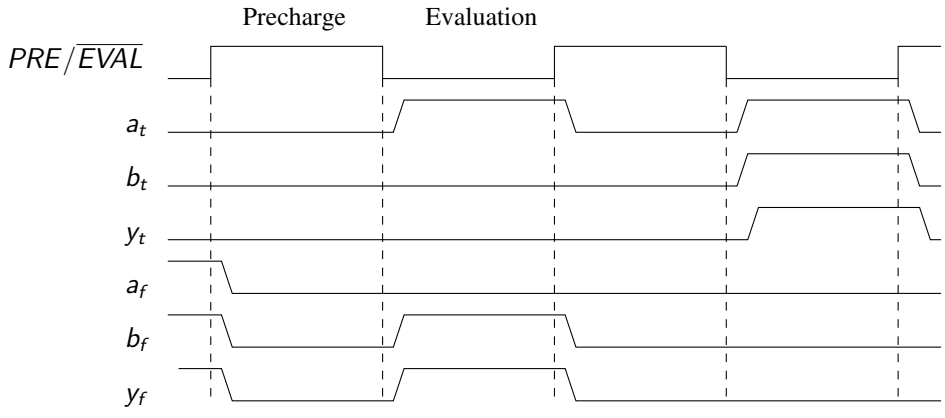
Plan

DPL overview



2 Networks: T and F



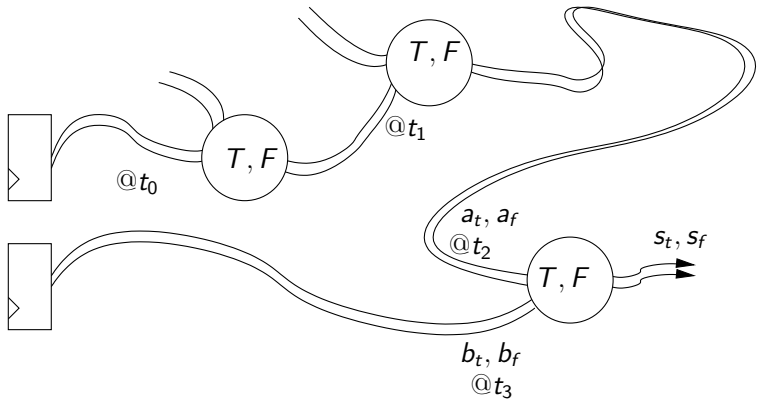


2 phases

DPL Drawbacks

Early Evaluation

The delay between two DPL inputs is observable at the output



Technological Biais

- OR consumption $\neq$ AND consumption
- routing T $\neq$ routing F

Solutions 1/2

Logic without glitches and early propagation

$\Rightarrow$ **Synchronization**

The rules to be "synchronized":

- **Rule 1:** Evaluation starts after all the input signals are valid.
- **Rule 2:** Precharge starts:
  1. Either after all the inputs becomes NULL (NULL is the value in precharge phase) but
     the outputs need to be memorized.
  2. Or before the first input becomes NULL
     which does not need any memorization.

Solutions 2/2

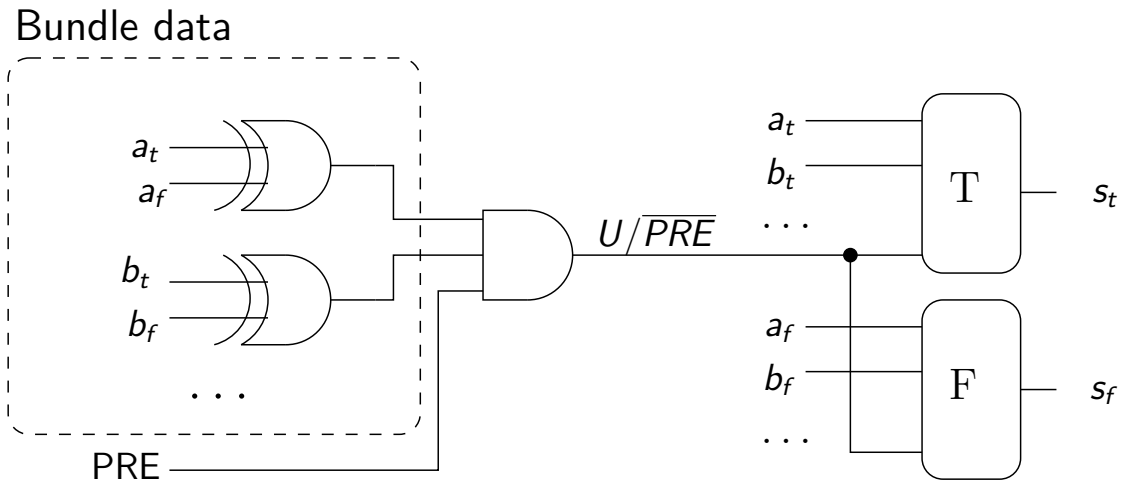
Logic with a minimum of technological biais

- Special care at placing and routing (but the FPGA vendors give few informations)
- Use of the same logic structure for True and False (e.g. MDPL with majority gates)
- Statistical balancing

Logic resistant to fault attacks

- Detection capability or
- Resilience

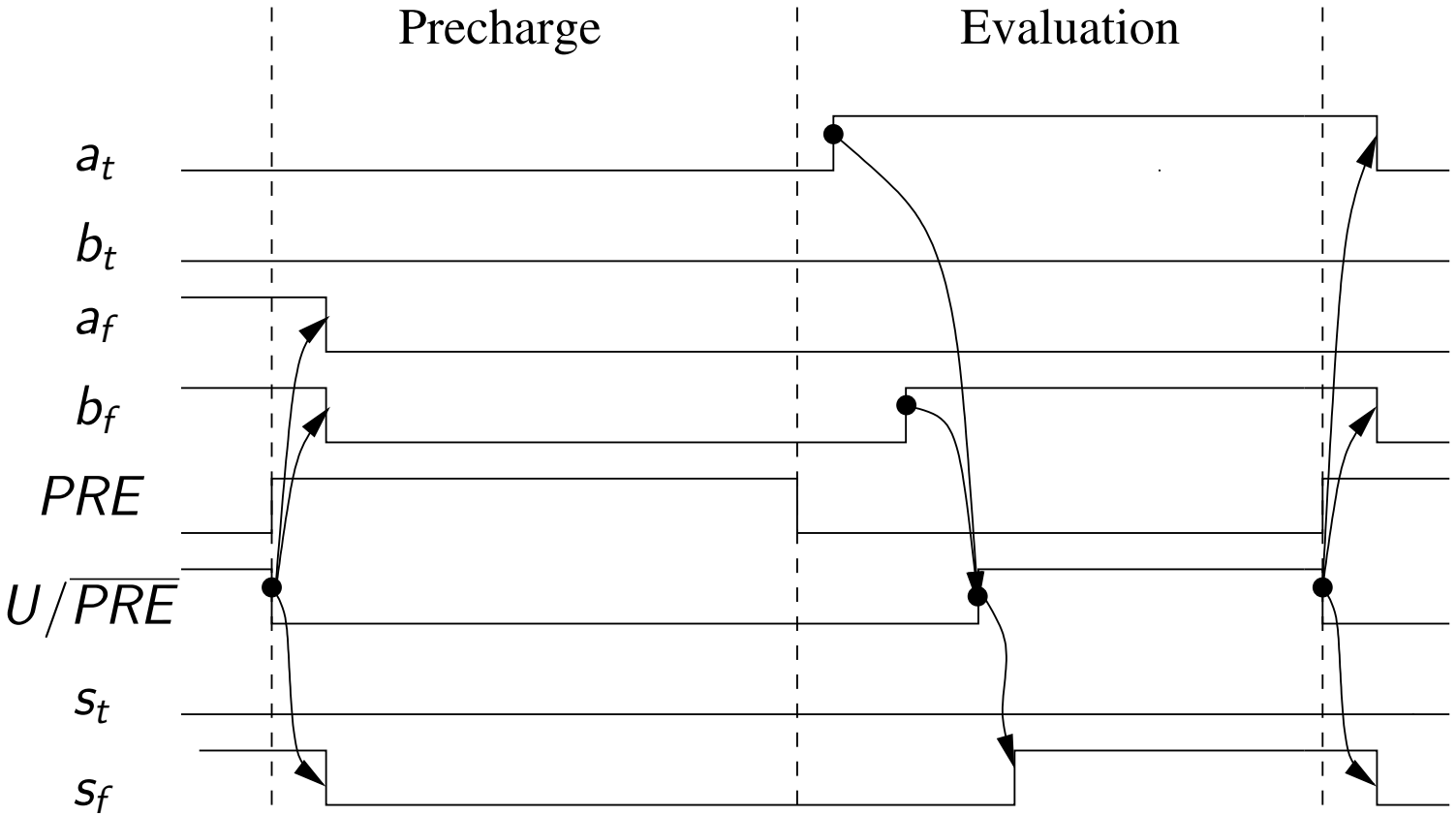
BCDL gate: Synchronization with Global Precharge



Global precharge *PRE* and "unanimity to 1"

- The global precharge replaces the "unanimity to 0".
- No need to memorization as *PRE* is faster than any inputs.
- $U/\overline{PRE}$ falls to 0 $\Rightarrow$ precharge is forced immediately.
- $U/\overline{PRE}$ raises to 1 $\Rightarrow$ evaluation begins after the last signal.

Exemple of a 2-input OR gate

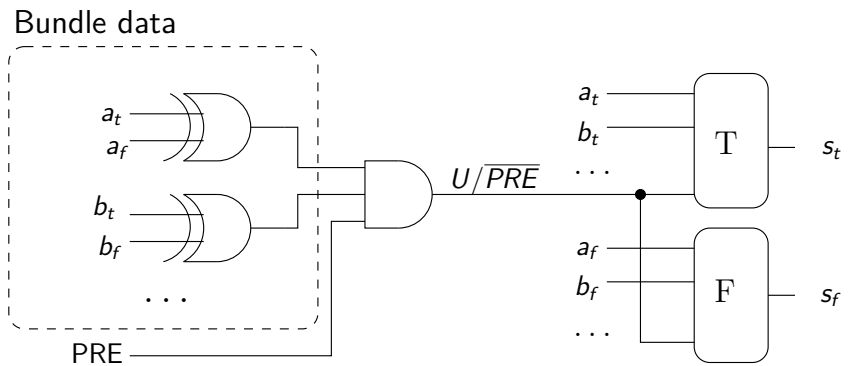


Robustness against FA

In-Built Robustness against Fault Attacks

- Automatically detects symmetric faults: {VALID0, VALID1} $\overset{\downarrow \text{ or } \uparrow}{\longrightarrow}$ {NULL0, NULL1}($1 \rightarrow 0$ or $0 \rightarrow 1$).
- "Error state" is propagated throughout the design $\Rightarrow$ **Fault resilience**.

PRECHARGE	Fault detection
1	state $\neq$ {NULL0, NULL1}
0	state$\neq$ {VALID0, VALID1}



Area

T and F easy to implement

- Not limited to positive functions
- **separable**
  - 1 additionnal input $(U/\overline{PRE})$ + duplication(T and F)
  - Area of tables $= 2.2^{n+1} < 2^{2n}$ if $n > 2$
  - $\Rightarrow$ S-Box area = only **4 times** the size of an unprotected one.

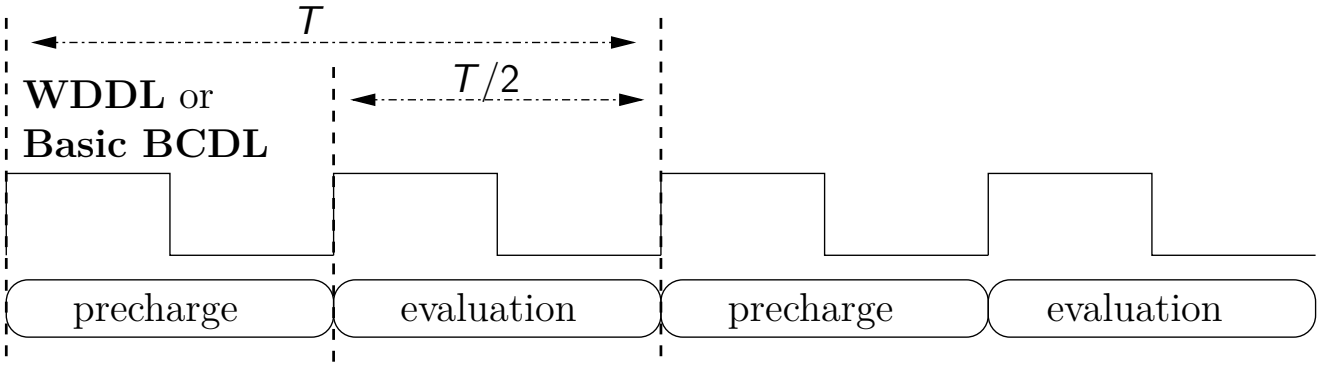
Total Area

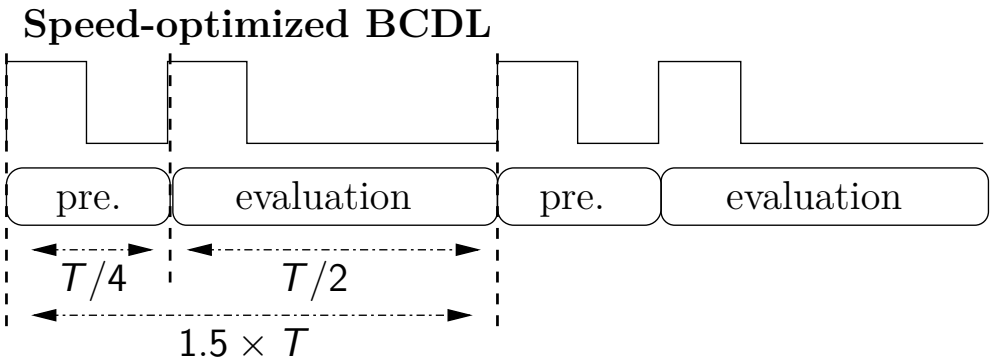
$=$ DFF$(*4)$ + [SYNC(*a few gates*) + T + F] $* n$.

Special case: MUX driven by single rail signal

- No needs of synchronization.

Speed optimization





Faster than other DPLs

- Evaluation time $>$ precharge time $\Rightarrow$ performances $\nearrow$
- Speed / $\sim 1.25 \leftrightarrow 1.75$

Comparison with other DPLs

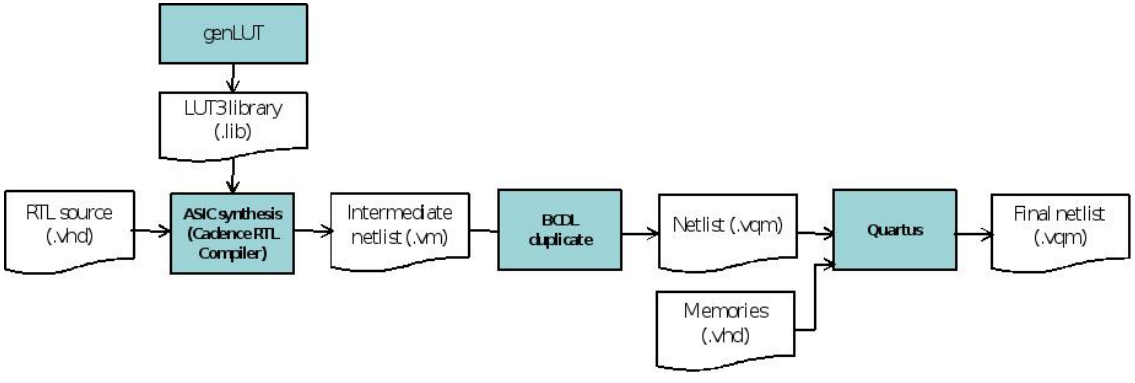
Logic	AND2 compl.	Speed	Robust. SCA		Robust. FA		Design Constr.
			EE	T. B.	Fault	Det.	
WDDL	2	$< 1/2$			asym	comb	Positive gates
STTL	15	$< 1/4$	✓		sym	seq	50% more wiring
Seclib	18	$< 1/2$	✓	✓	sym	seq	custom cell
IWDDL	14	$< 1/2 \cdot n$	✓		asym	com	superpipeline
BCDL	8	$> 1/2$	✓		sym	comb	
MDPL	10	$< 1/2$		✓	asym	comb	MAJ gate + RNG
iMDPL	27	$< 1/2$	✓	✓	asym	comb	MAJ gate + RNG
DRSL	17	$< 1/2$	partly	✓	sym	comb	+ RNG
MBCDL	20	$> 1/2$	✓	✓	sym	comb	+ RNG

Plan

How to design with BCDL ?

Two methods

- Automatic: Top-Down



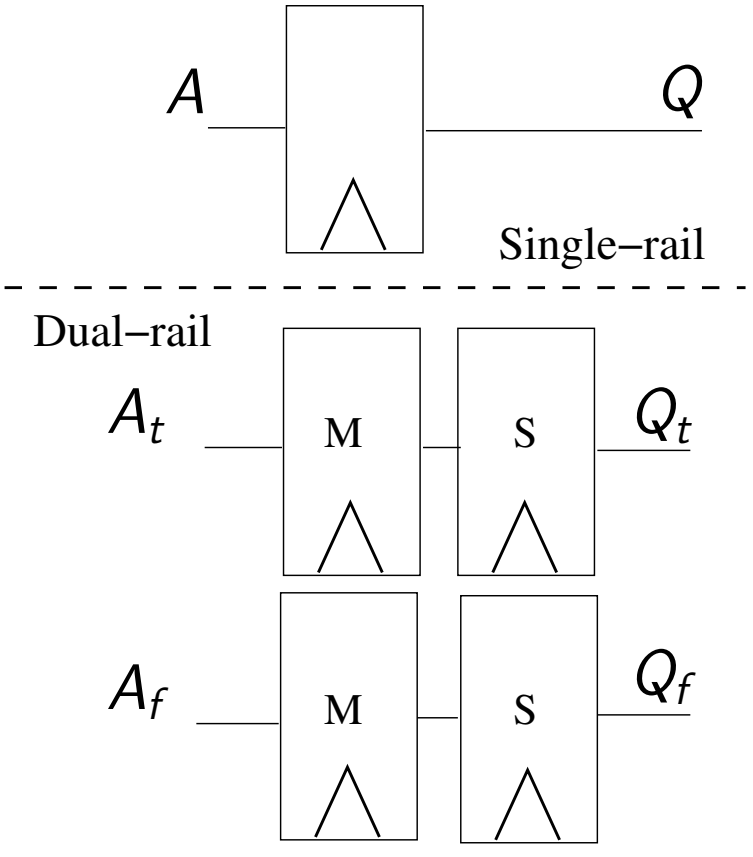
- Manual: Bottom-up
  - Based on primitives
  - Could be Optimal in robustness, complexity, speed but
  - Design constraints

Bottom-Up Approach to design AES in BCDL

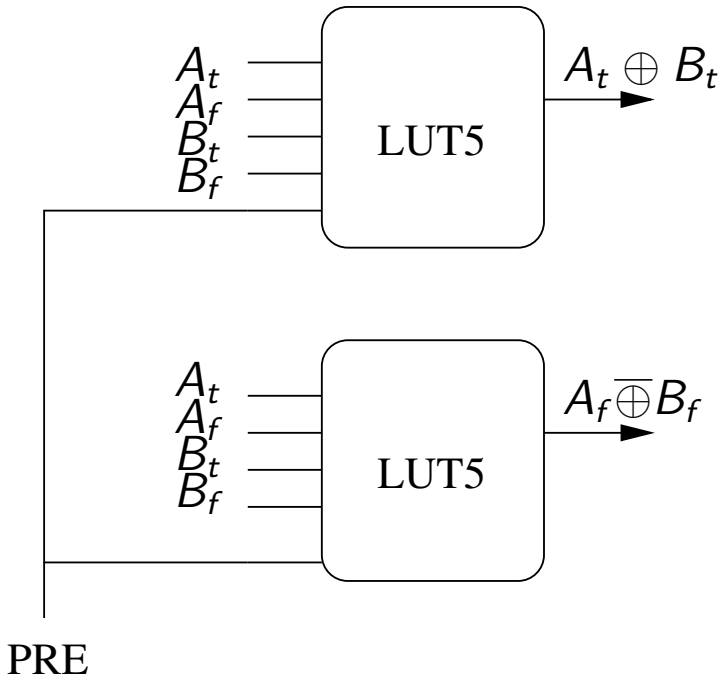
Basic Steps

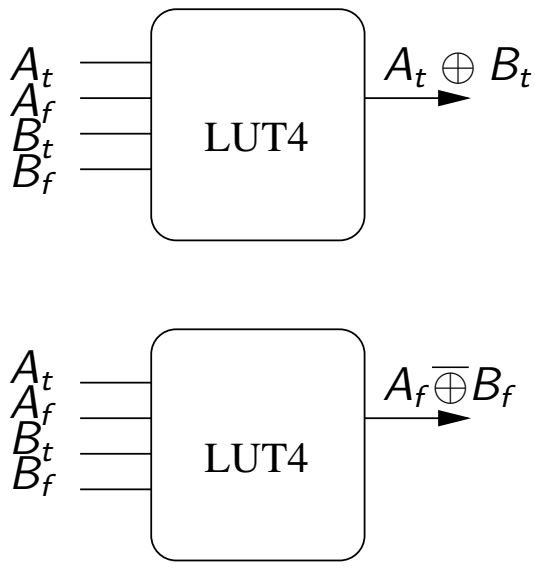
- Write a structural RTL code for singlerail AES using just the primitives.
- 4 primitives in AES $\Rightarrow$ DFF, MUX, RAM, XOR.
- Duplicate all the data signals, control signals are left single rail.
- Replace the identified primitives by dual-rail BCDL compliant primitives.
- A wrapper connects single rail I/O's to dual rail data signals and ensure two phase operation of I/O's.
- The FSM should work at half its nominal frequency.

BCDL DFF



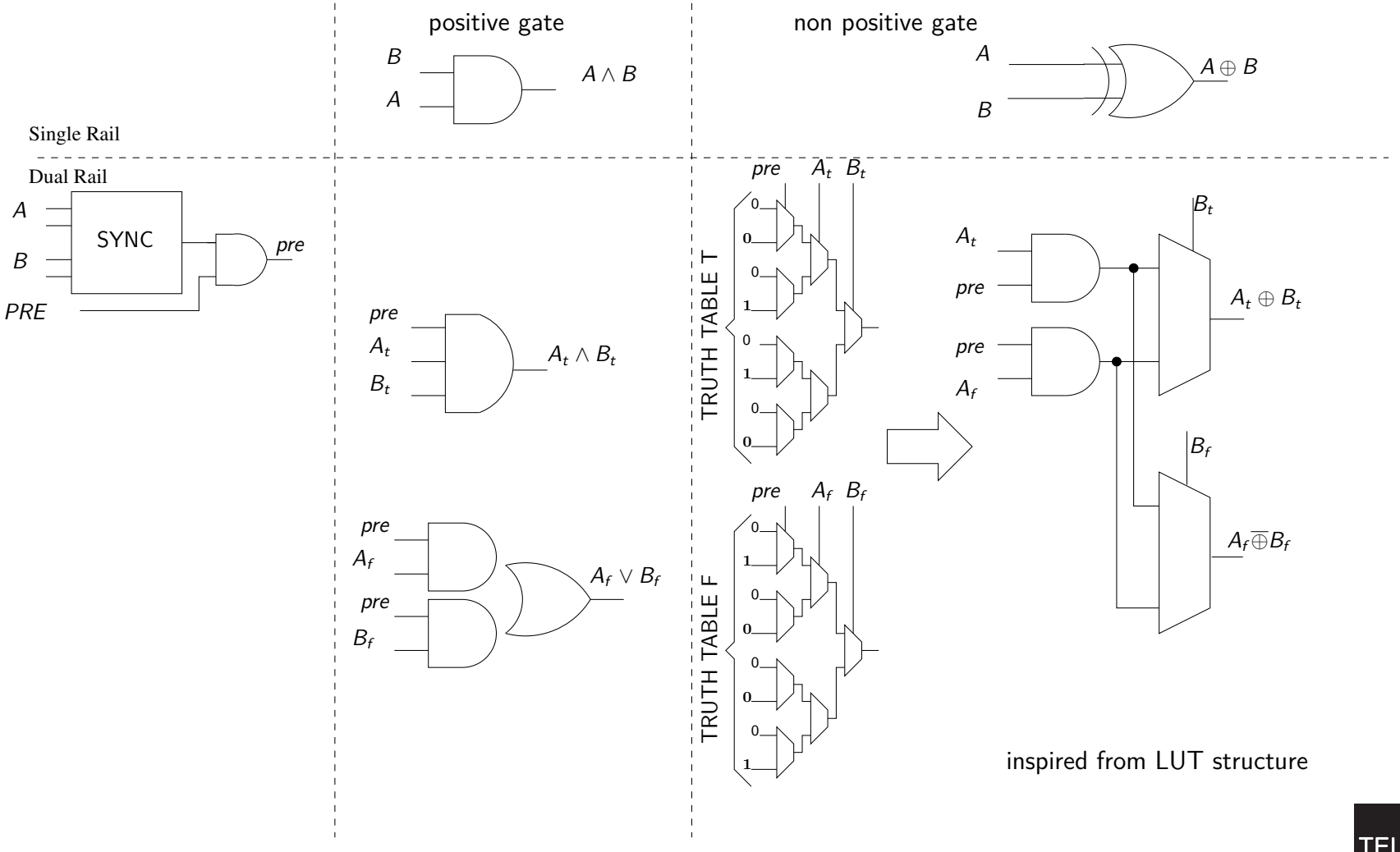
BCDL 2-input gates in FPGA



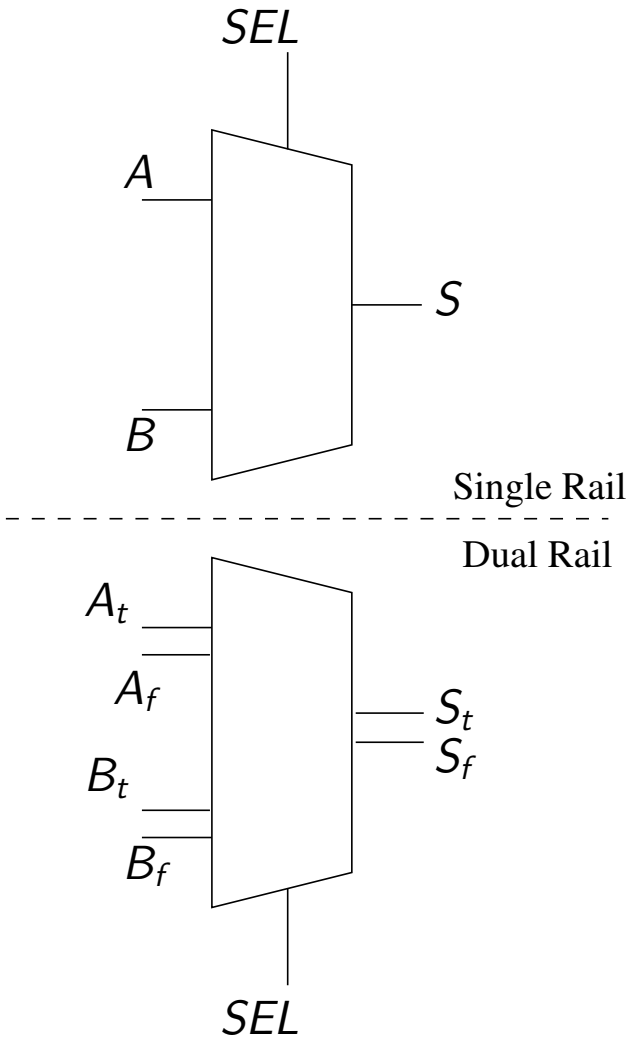


PRE is necessary for Hi-speed architectures, is the latest input of LUT

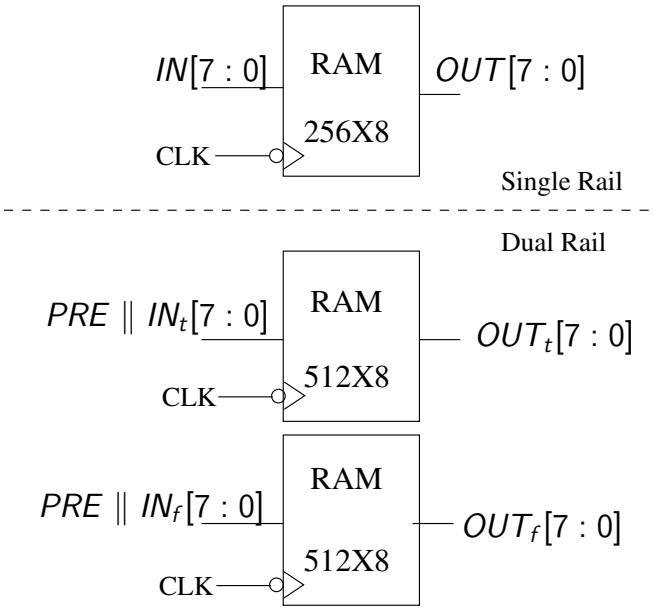
BCDL gates in ASIC



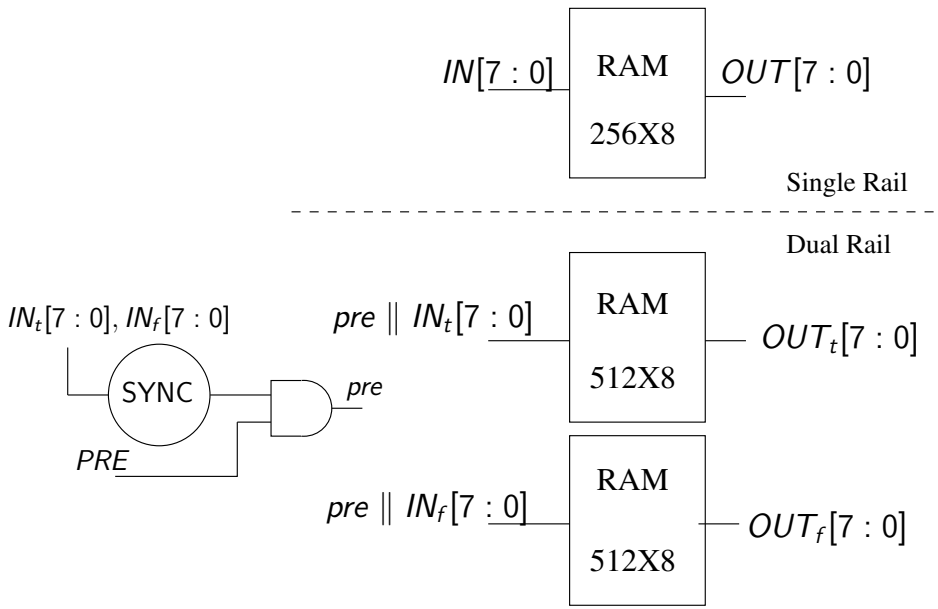
BCDL MUXes



BCDL RAM and/or large gates

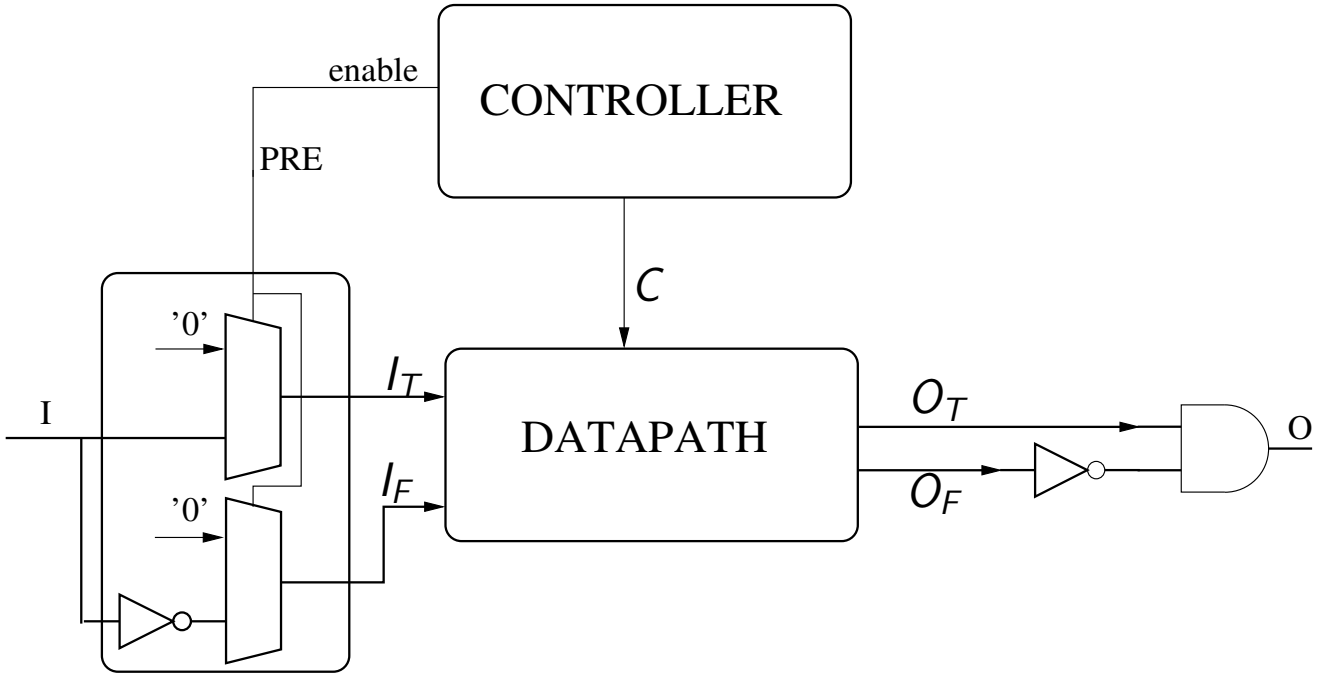


Synchronous RAM



Asynchronous RAM

BCDL Wrapper

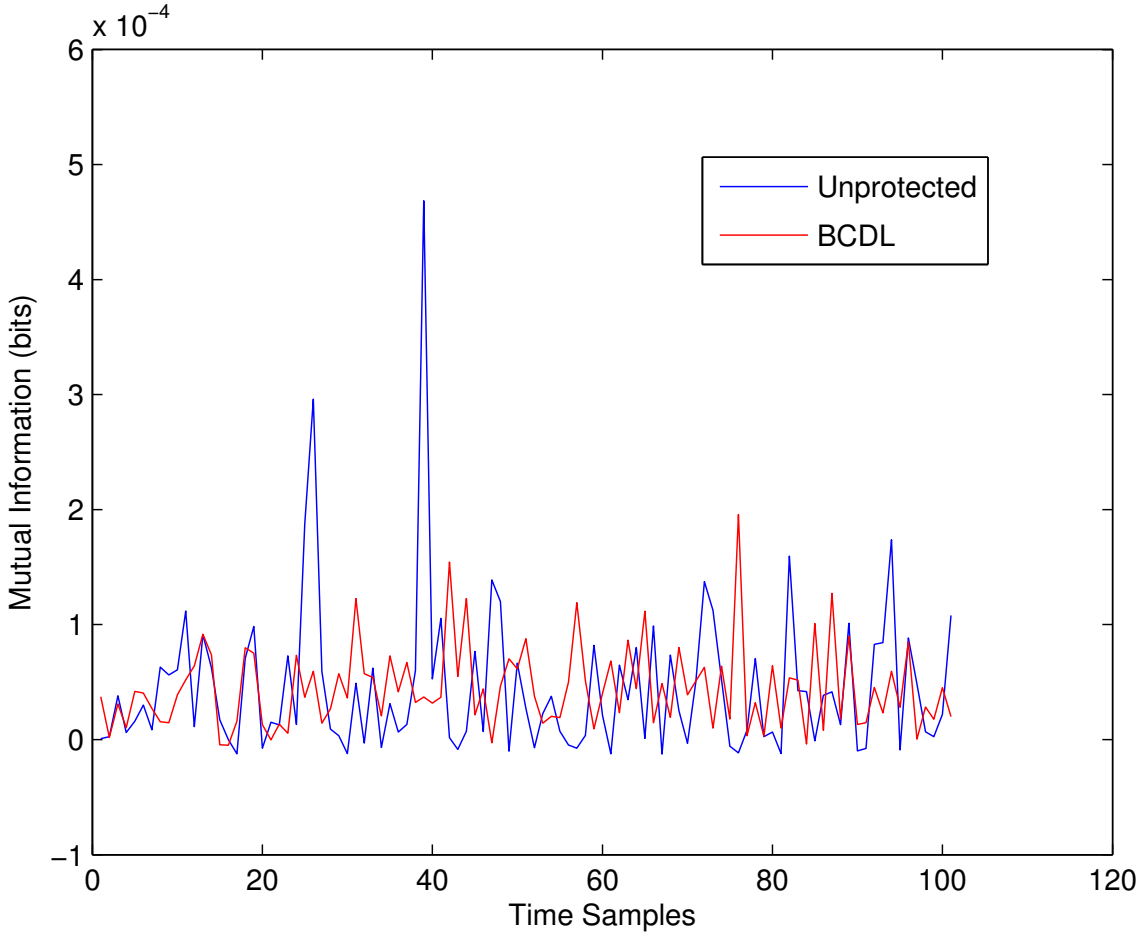


- All global inputs converted to true and false representation.
- Signals from/to other modules converted to dual-rail/single-rail.
- Signal **Phase** forces '0' on all the inputs during precharge and freezes the controller to its current state.

Complexity and speed results of AES

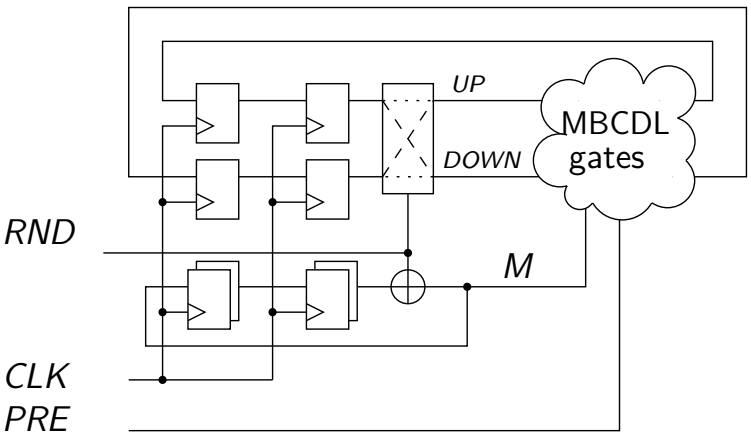
Design	Cost	LUT type	Max. Frequency
REF	2396	LUT4	56.9 MHz
WDDL	12530	LUT4	21.3 MHz
DPL-noEE	14126	LUT4	19.7 MHz
REF-BCDL	792	LUT5/LUT6	122.7MHz
BCDL	2406	LUT5/LUT6	73.6 MHz

Robustness analysis

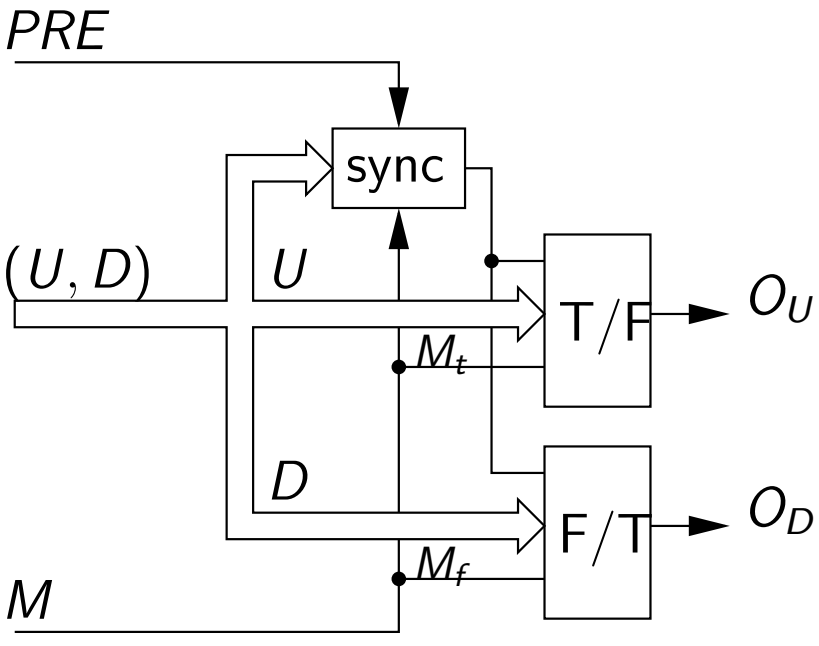


Mutual information for SBOX 0, bit 2

Mask-BCDL to mitigate the P/R constraint



Global architecture



MBCDL gate

Plan

BCDL implementation : Summary

- Automatic: Top-down, sub optimal
- Manual: Bottom-Up method Based on primitives
- Less complex and fast
- ASIC gates inspired from FPGA LUTs
- Takes advantage of ROM (as separated T and F networks)
- Less sensitive to routing congestion
- First tests show excellent robustness
- Design constraint : primitives placement in an atomic manner
- P/R improvable by Mask-BCDL