Tolerating Multiple Faults With Proximate Manifestations in FPGA-Based Critical Designs For Harsh Environments

Jaime Espinosa, David de Andrés, Juan-Carlos Ruiz and Pedro Gil
{**jaiesgar**, ddandres, jcruizg, pgil}@disca.upv.es

Oslo, NORWAY
August 2012

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

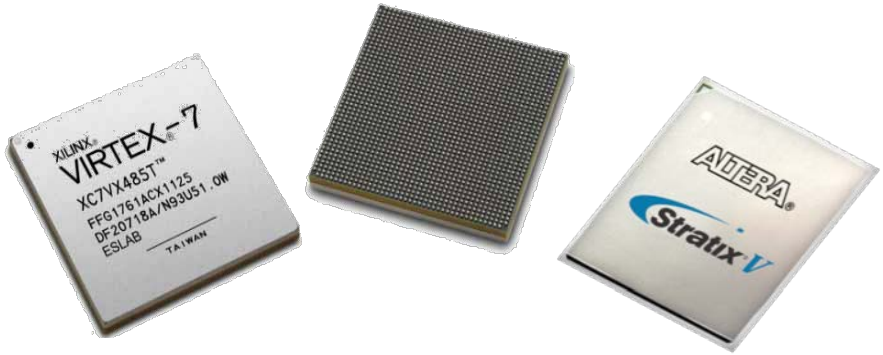
FPL2012 22nd International Conference on Field Programmable Logic and Applications

ITACA GSTF Fault-Tolerant Systems Group
Institute for the Applications of Advanced ICTs

The scenario

- Traditional FPGA products for harsh environments are dependable … and costly.

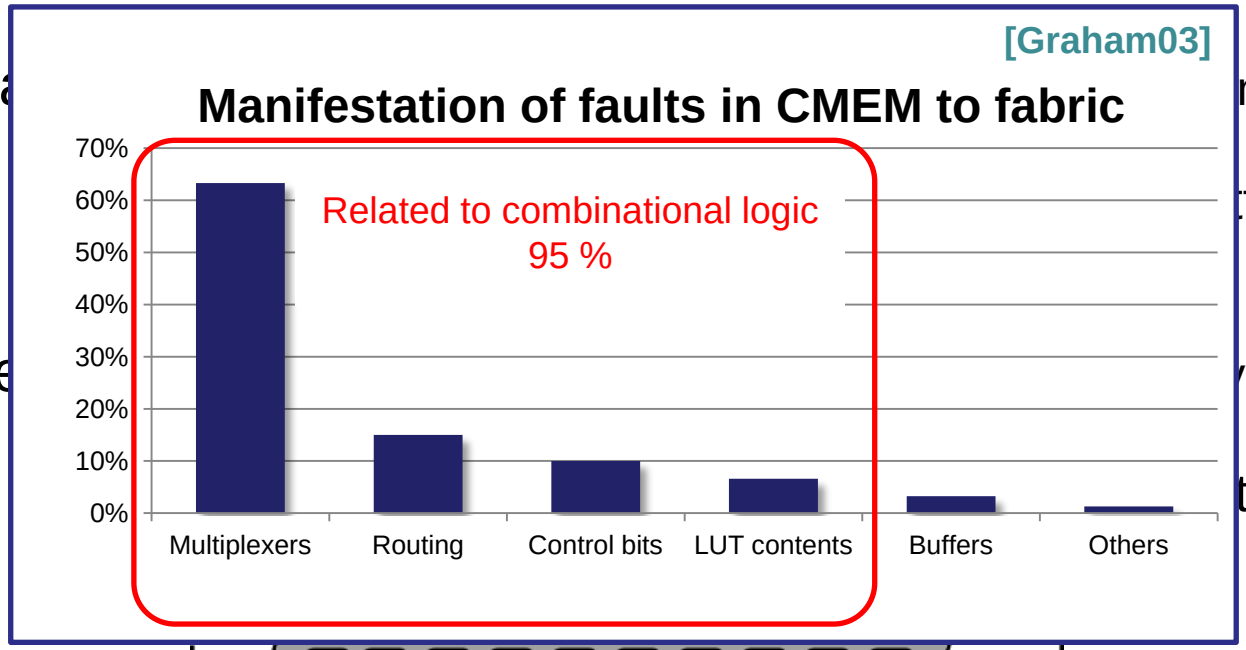
- New SRAM-based devices are low-cost, powerful and flexible… but inherently not so dependable.



SRAM FPGAs dependability impairments

- Fa… …nditions.
- Te… …tent
- Te… …, lower voltage
- …te in time
- Architecture: Fabric <1% sequential logic, CMEM faults manifest in fabric
  - Faults mostly affect combinational logic

[Graham03]



Protect combinational logic of designs against the occurrence of multiple (transient/permanent) faults

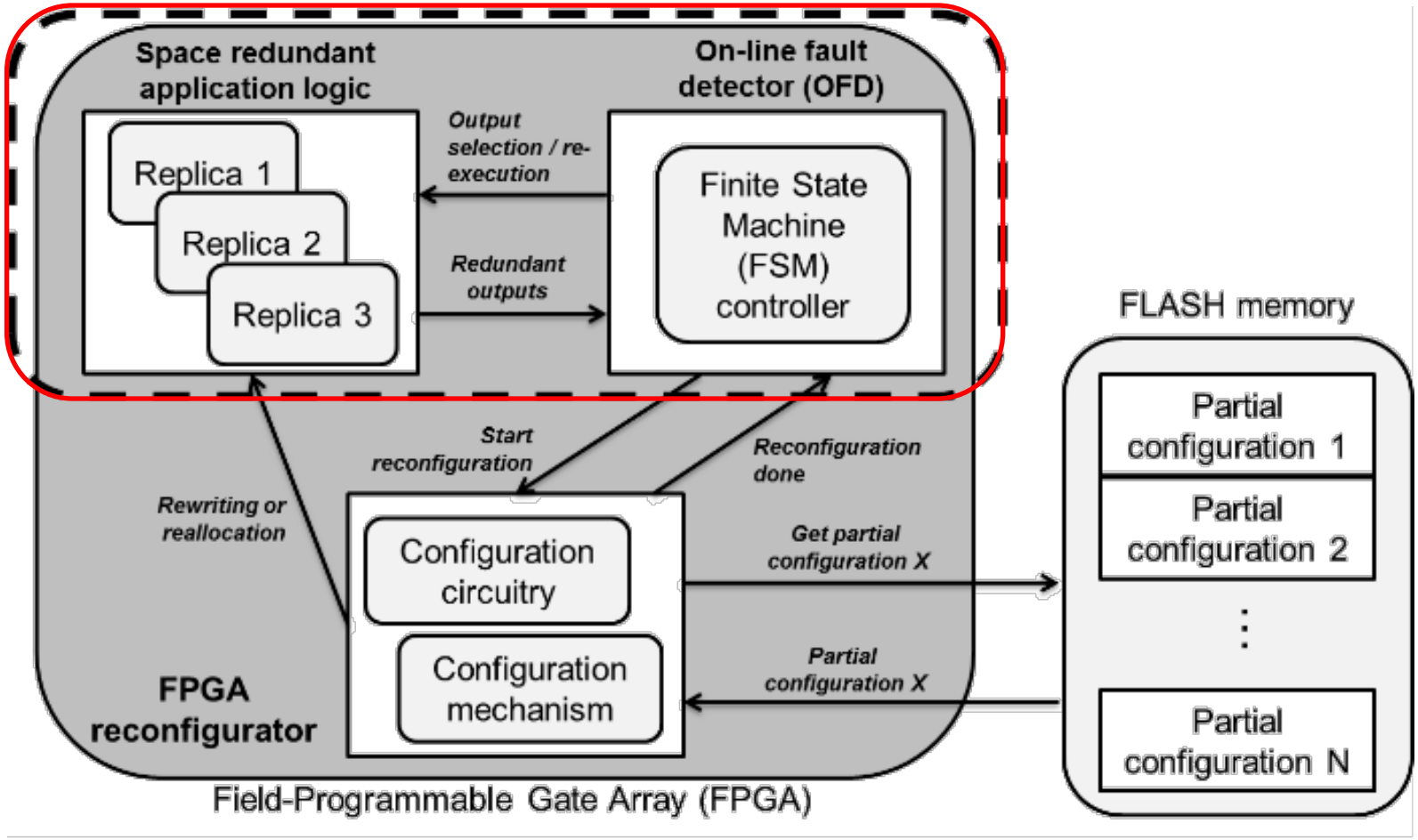
Existing g

Not one is solution to multiple (transient/permanent) faults with proximate manifestations

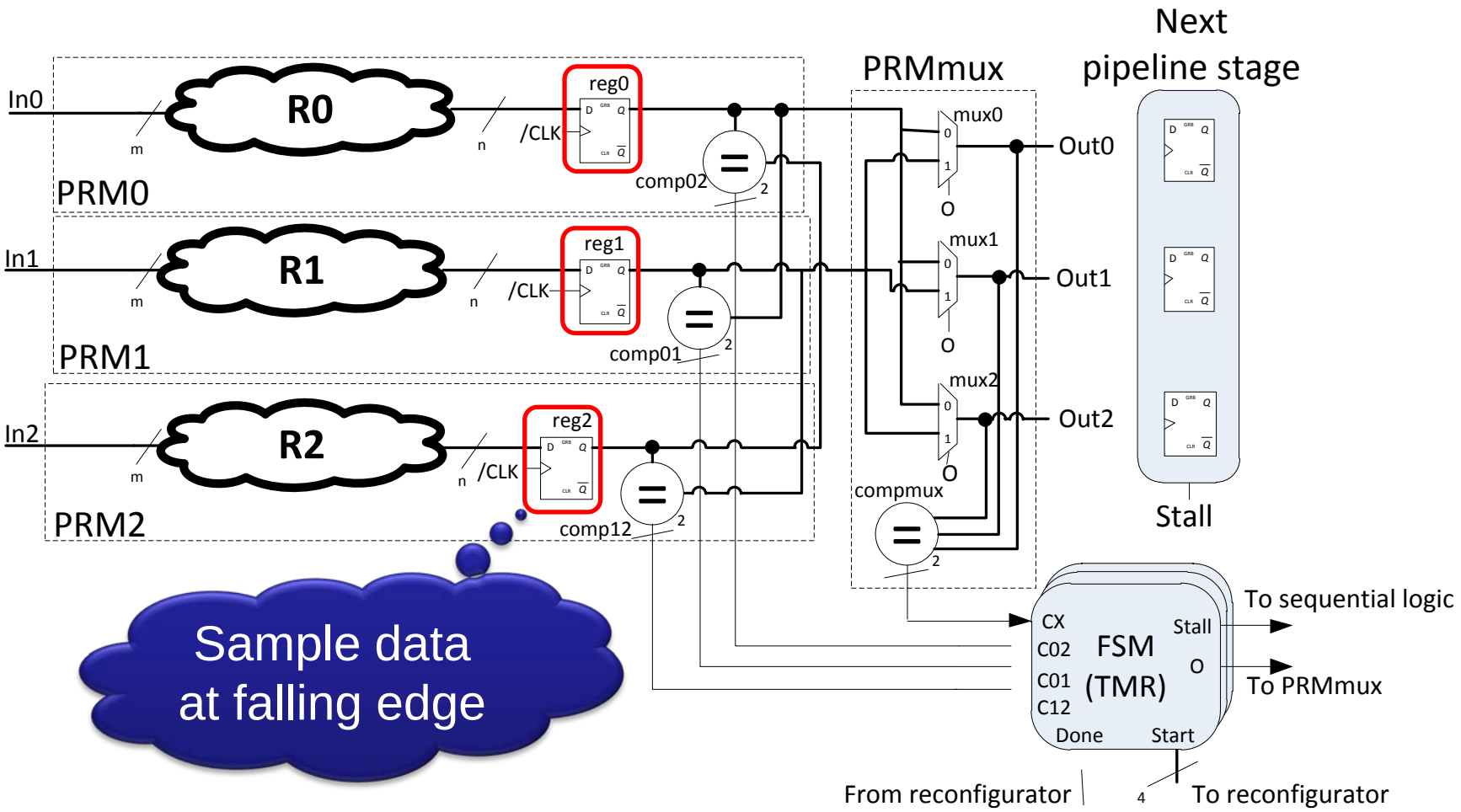
	Technique	Handled cases
Detection and masking	Triple Modular Redundancy	Single transients & permanents in fabric Multiple transients in fabric
	Time Redundancy	Single transients in fabric
	Dual Redundancy + Time redundancy	Single transients & permanents in fabric

	Technique	Handled cases
Recovery	Scrubbing	Transients in CMEM
	Direct Rewriting (Partial Dyn. Reconf.)	Transients in CMEM
	Relocating	Multiple permanents in fabric & CMEM

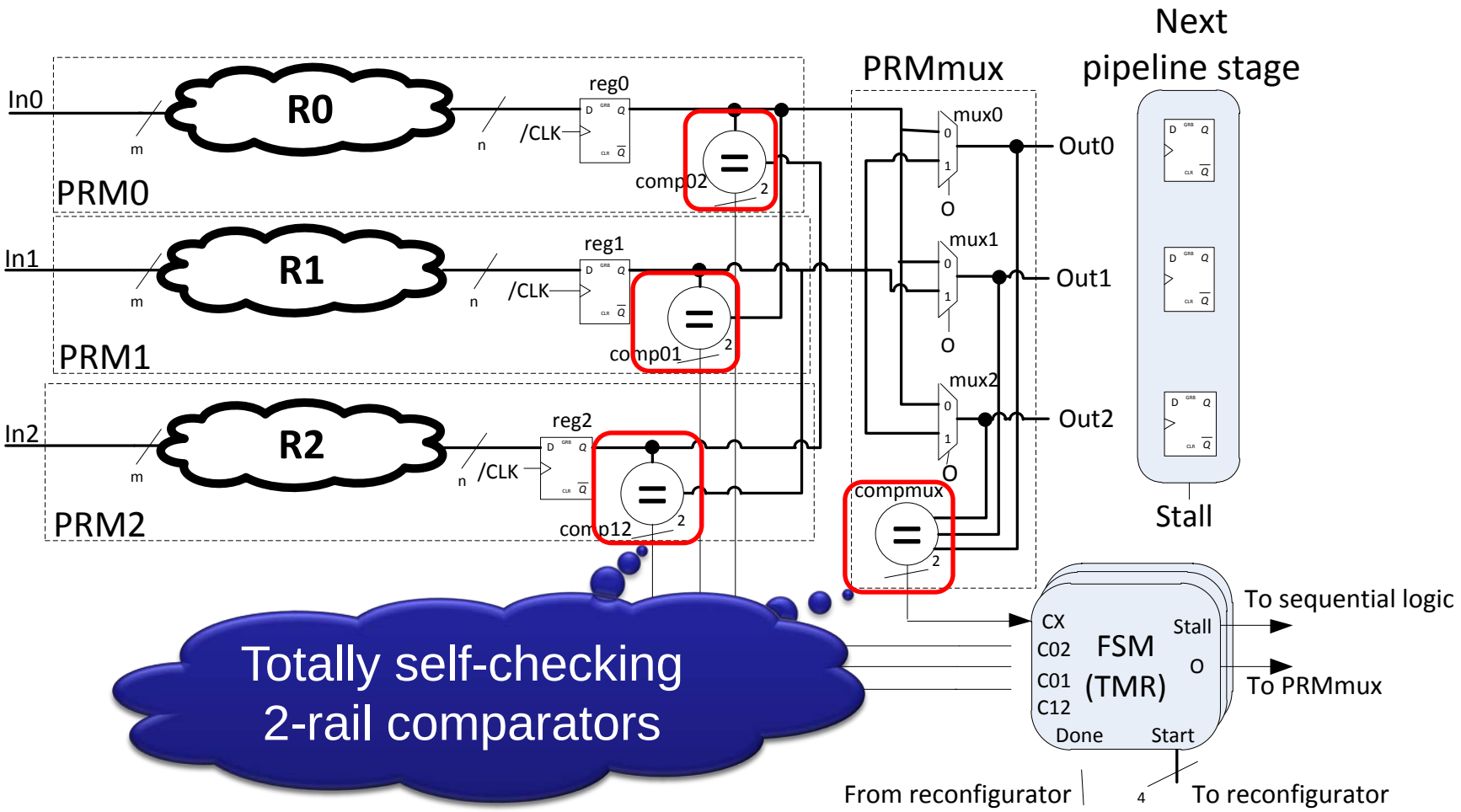
Our approach to tolerate MF: TMR-MDR
(Triple Modular Redundancy - Module Discard and Repair)



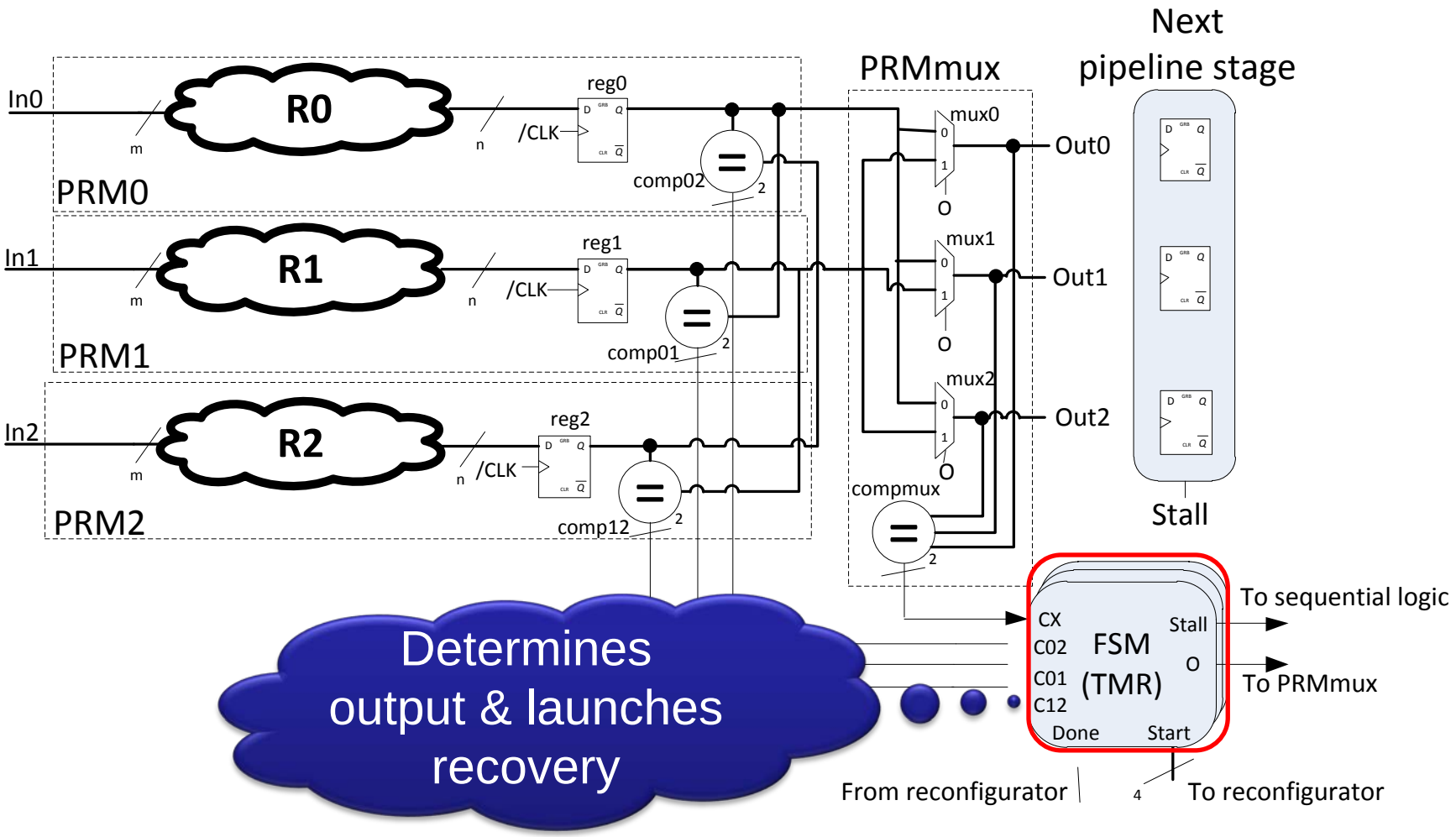
Detailed architecture (I)



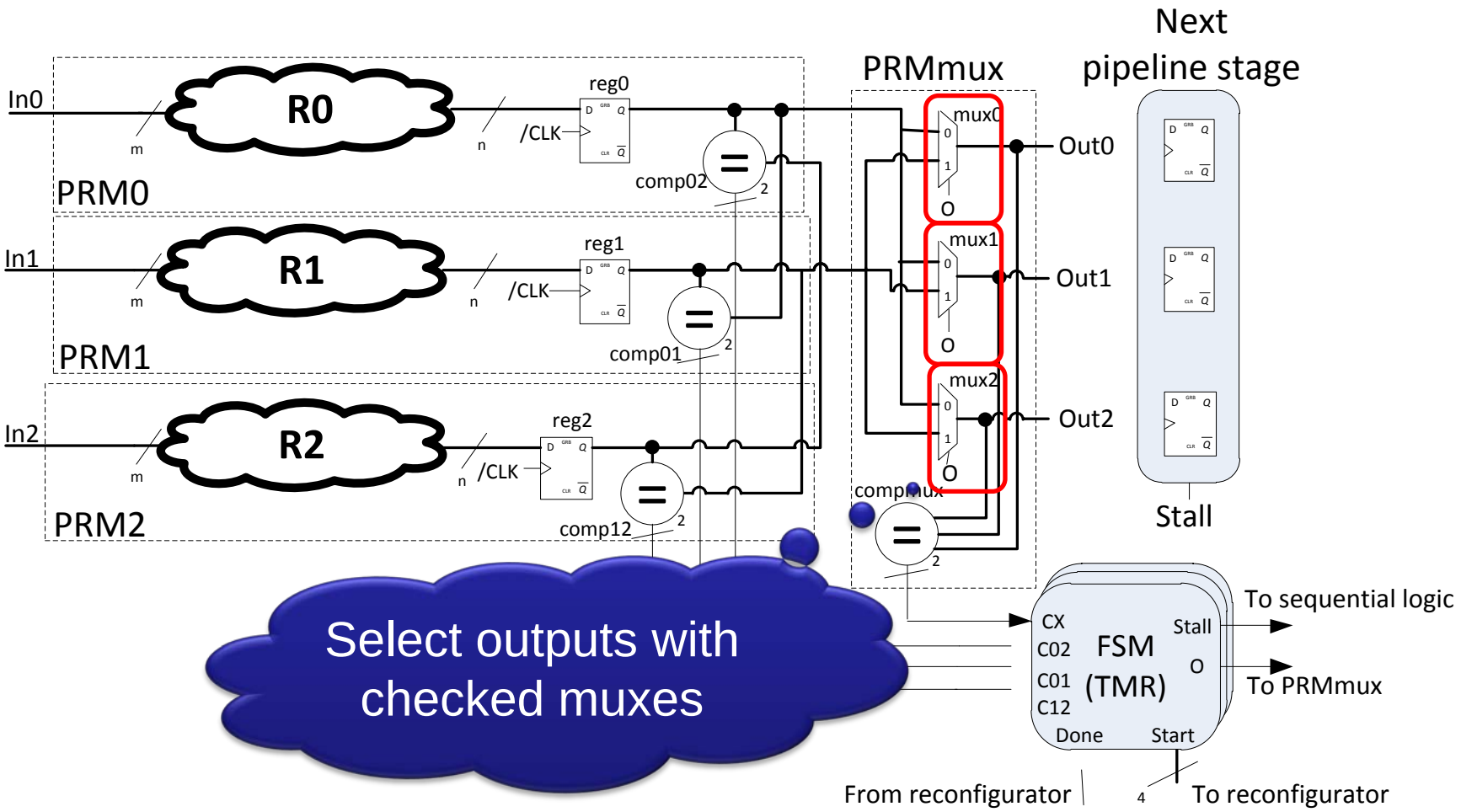
Detailed architecture (II)



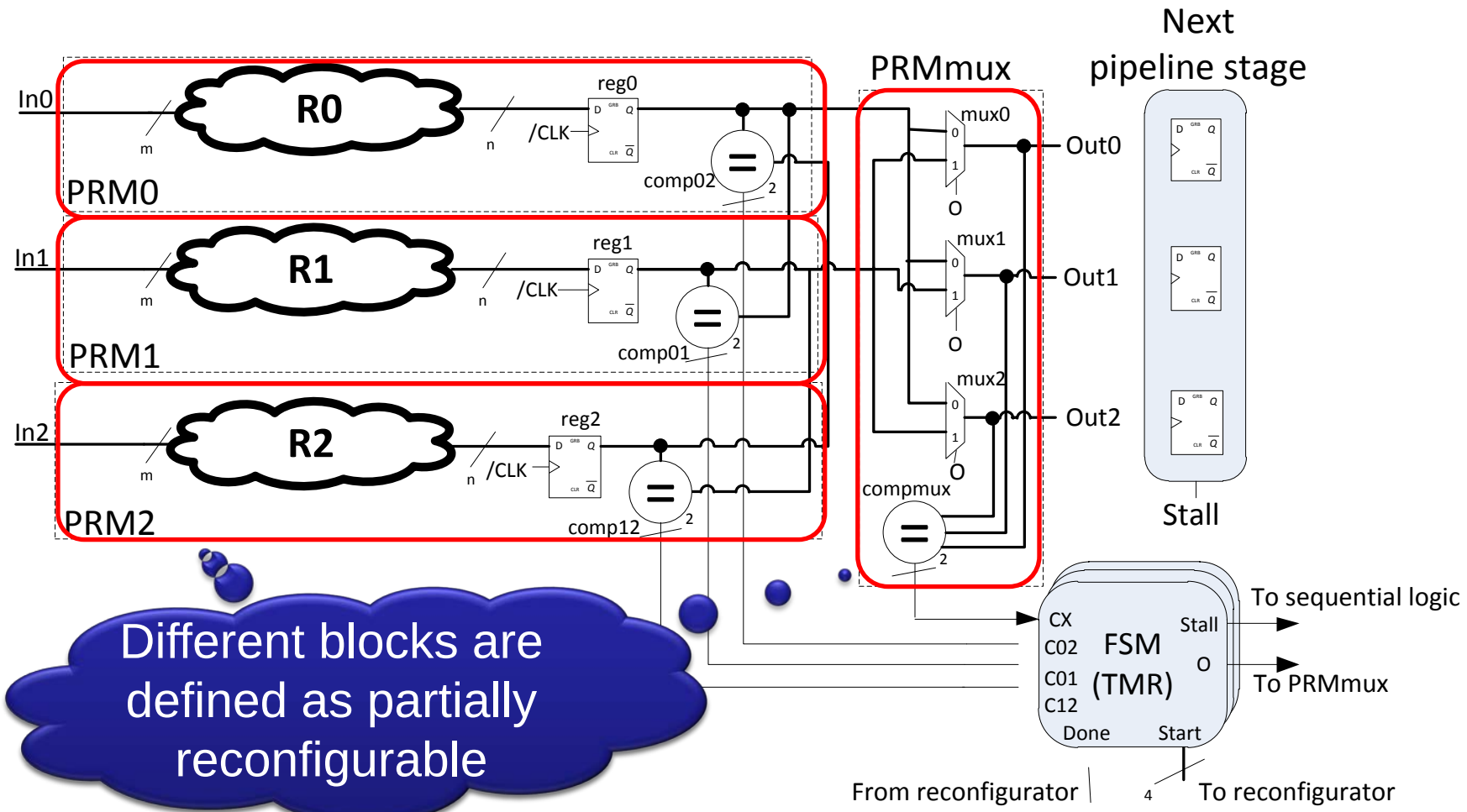
Detailed architecture (III)



Detailed architecture (IV)



Detailed architecture (V)

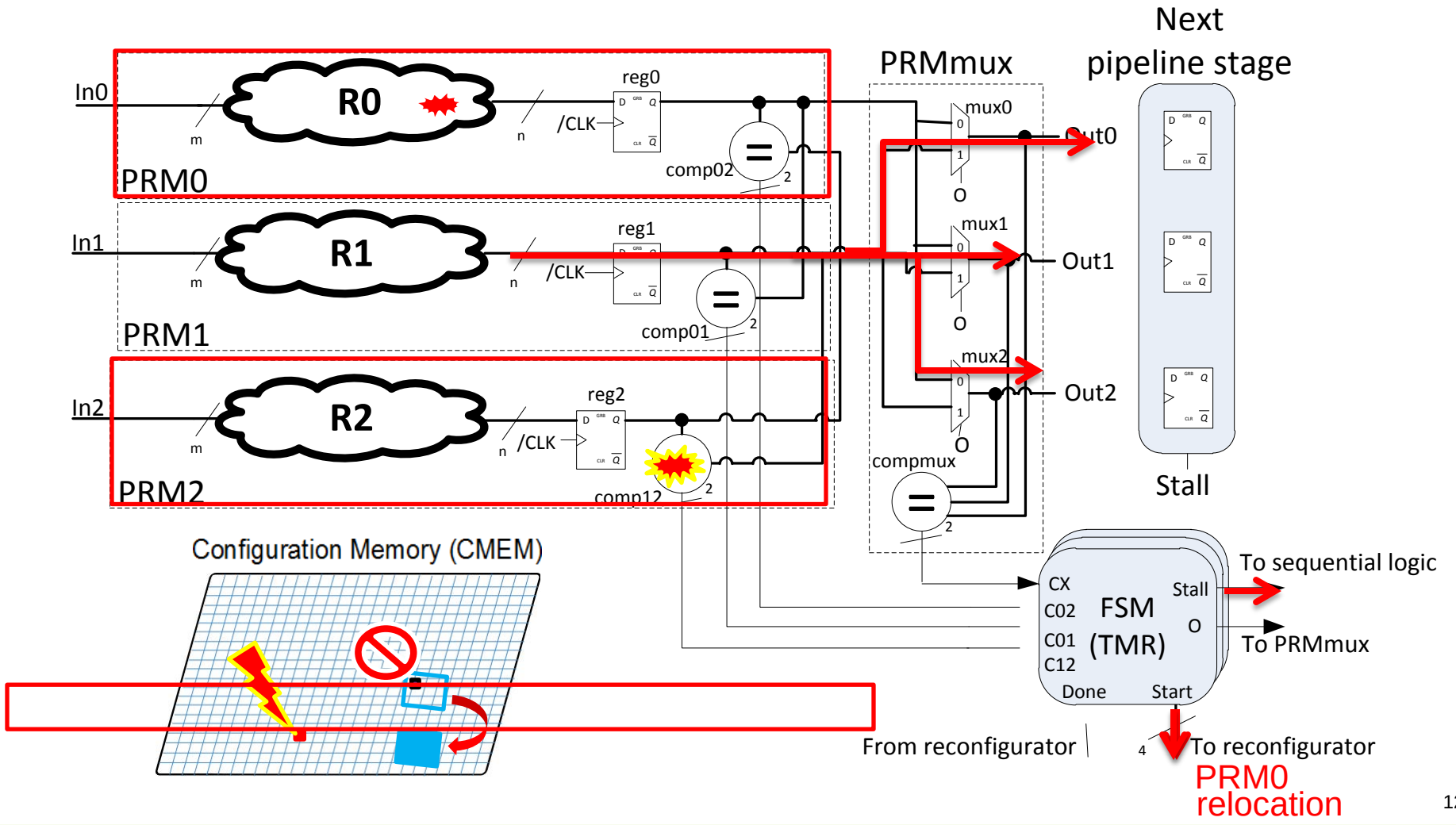


FSM Design: outputs

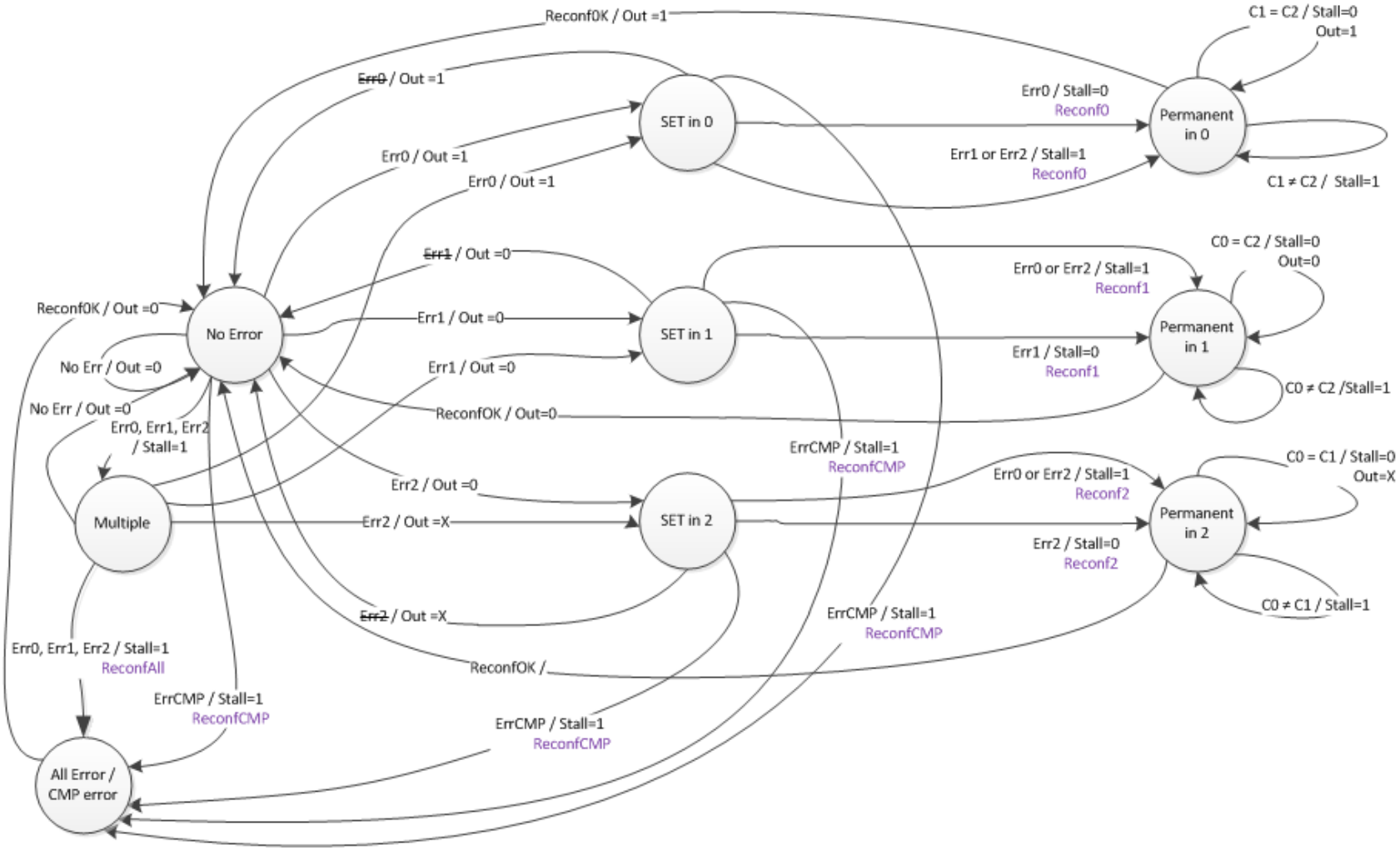
Recovery Action
No action
Rewriting
Relocation
Multiple Rewriting/relocation

Control Flow
No stall
Single stall
Multiple stall

FSM Design: case examples



FSM Design: state diagram



Case study

- **Target design**: 32-bit floating point multiplier
  - Extensive use of combinational logic
  - Extensive use of internal routing
- **Implementations** [target: Virtex-6 (XC6VLX240T-1FFG1156)]:
  - Base (32-bit FP multiplier)
  - TMR enhanced with triple majority voter (eTMR)
  - TMR with Module Discard and Repair (TMR-MDR)
- **Compared features of implementation**:
  - Resource utilisation (silicon area)
  - Performance (maximum clock frequency)
- **Compared measures**:
  - Percentage of failures
  - Percentage of experiments leading to stall

Testing campaign

- VHDL-based Fault Injection Tool (VFIT).
- **Workload**: large set of randomly generated input operands.
- **Faultload**: 6 different configurations related to considered fault models (multiple faults represented with 2 consecutive faults)

Table 1. Considered single fault models

Target	Duration	Manifestation on fabric
Combinational logic (fabric)	Transient	Pulse, indetermination, and delay
	Permanent	Stuck-at, stuck-open, indetermination, delay, short, open, and bridging
Sequential logic (CMEM & fabric)	Transient	*Transient* stuck-at, bit-flip, indetermination, delay, short, and open
	Permanent	*Permanent* stuck-at, indetermination, delay, short, and open

Table 2. Considered multiple fault models

1st fault		2nd fault	
Duration	Target	Duration	Target
Transient	Comb. (fabric)	Transient	Comb. (fabric)
Transient	Comb. (fabric)	Transient	CMEM [1]
Transient	Comb. (fabric)	Permanent	Comb. (fabric) or CMEM
Transient	CMEM [1]	Transient	Comb. (fabric)
Permanent	Comb. (fabric) or CMEM	Transient	Comb. (fabric)
Transient	CMEM [1]	Transient	CMEM [1]
Permanent	Comb. (fabric) or CMEM	Permanent	Comb. (fabric) or CMEM

[1] Transient faults in CMEM manifest as permanent ones in design logic (fabric) and can be assimilated to them from the logic point of view.

- Total number of experiments (250800) related to number of elements in each circuit.

Results: failures percentage

Faultload[1]	Enhanced TMR			TMR-MDR approach		
	Number of experiments	Number of failures		Number of experiments	Number of failures	
T	18237	0	(0.00%)	23563	0	(0.00%)
P	18237	0	(0.00%)	2	0	(0.00%)
T + T	18237	83	(0.46%)		2	(0.01%)
T + P	18237	113	(0.62%)		2	(0.01%)
P + T	18237	3271	(17.94%)	23563	22	(0.09%)
P + P	18237	3085	(16.92%)	23563	22	(0.09%)
Total	109422	6552	(5.99%)	141378	48	(0.03%)

0.61 PP

17.85 PP

16.83 PP

Results: temporal intrusion

Faultload[1]	Number of experiments	Experiments leading to single stalls		Experiments leading to multiple stalls *	
T	23563	0	(0.00%)	2402	(10.19%)
P	23563	0	(0.00%)	3002	(12.74%)
T + T	23563	2084	(8.84%)	4849	(20.58%)
T + P	23563	115	(0.49%)	7387	(31.35%)
P + T	23563	4214	(17.88%)	10416	(44.20%)
P + P	23563	12	(0.05%)	14623	(62.06%)
Total	141378	6425	(4.54%)	42679	(30.19%)

* A 32% of those experiments affected the muxes block, which accounts for just 5% of the occupied area

Results: Area and CK Period overhead

System	Area	Overhead	Clock period	Overhead
Original	231 CBs	—	25.613 ns	—
Enhanced TMR	820 CBs	254%	30.864 ns	21%
TMR-MDR approach	846 CBS	266%	36.164 ns	41%

Conclusions

- It is **possible** to use **SRAM-based** FPGAs in harsh environments
  - The use of TMR-MDR **improves coverage** of **multiple proximate faults** in time over existing techniques.
  - The **distinction between transient and permanent** faults allows for **reduced downtime** periods, thanks to the use of partial dynamic reconfiguration for fault tolerance and recovery.
  - **The cost is comparable to existing techniques**
- **Current research:** More accurate detection and diagnosis of faults → increased availability & improved management of resources

Thank you for your attention!

Any Questions?

Tolerating Multiple Faults With Proximate Manifestations in FPGA-Based Critical Designs For Harsh Environments

Jaime Espinosa, David de Andrés, Juan-Carlos Ruiz and Pedro Gil

{**jaiesgar**, ddandres, jcruizg, pgil}@disca.upv.es

Oslo, NORWAY

September 2011

FPL2012 *22nd International Conference on Field Programmable Logic and Applications*

Institute for the Applications of Advanced ICTs