VILLANOVA
UNIVERSITY
College of Engineering

A Separation and Protection Scheme for On-Chip Memory Blocks in FPGAs

Luis E. Ramírez Rivera, Xiaofang Wang, Danai Chasaki

FPGAs

- Field-Programmable Gate Arrays
- Configurable after manufacturing.
- Complex circuits can be designed.
- Lower frequency than most ASICs, but more flexible.

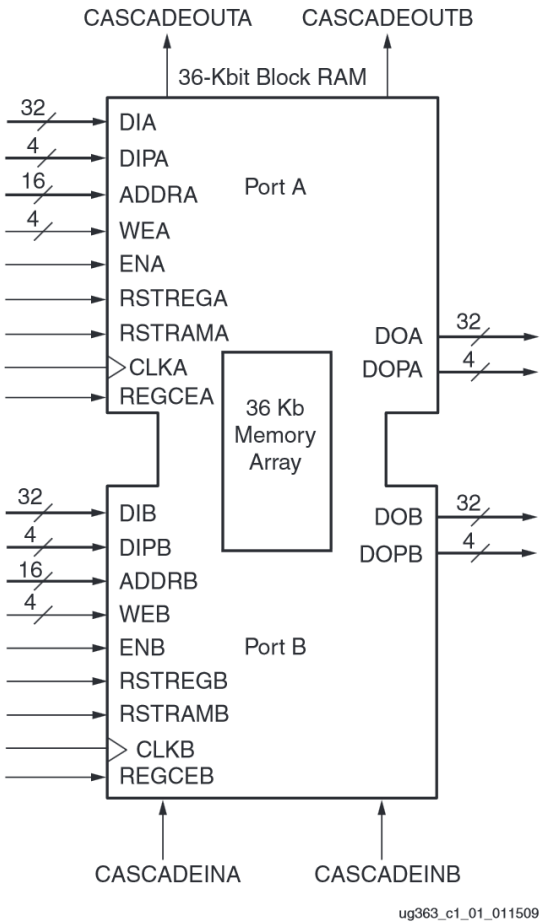
"Altera StratixIVGX FPGA" by Altera Corporation - Altera Corporation. Licensed under CC BY 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Altera_StratixIVGX_FPGA.jpg

Types of Threats

- Who
  - The competition
  - Black-hats (criminals)
  - The government
- What
  - Find keys
  - Steal bitstreams/IP
  - Insert Hardware Trojan
  - Learn sensitive data
  - Deny service
- Why
  - Steal IPs to use, sell or reverse engineer
  - Clone/Counterfeit
  - Circumvent security measures
  - Financial/identity fraud/theft
  - Wreak havoc (e.g., power grid)
- How
  - Technical attack

Typical Memory Types in FPGAs

- Off-Chip
  - SRAM
  - SDRAM
  - Flash
  - RLDRAM
- On-Chip
  - Simple Flip-flops
  - Block RAM
  - Distributed RAM



[5]

Problem

- On-chip memory is practically unprotected
  - Flat memory-addressing scheme & nothing else!
- It is also becoming more plentiful
  - More data will be vulnerable
- Cannot use encryption for security
  - Encryption takes a long time
  - Uses a lot of resources

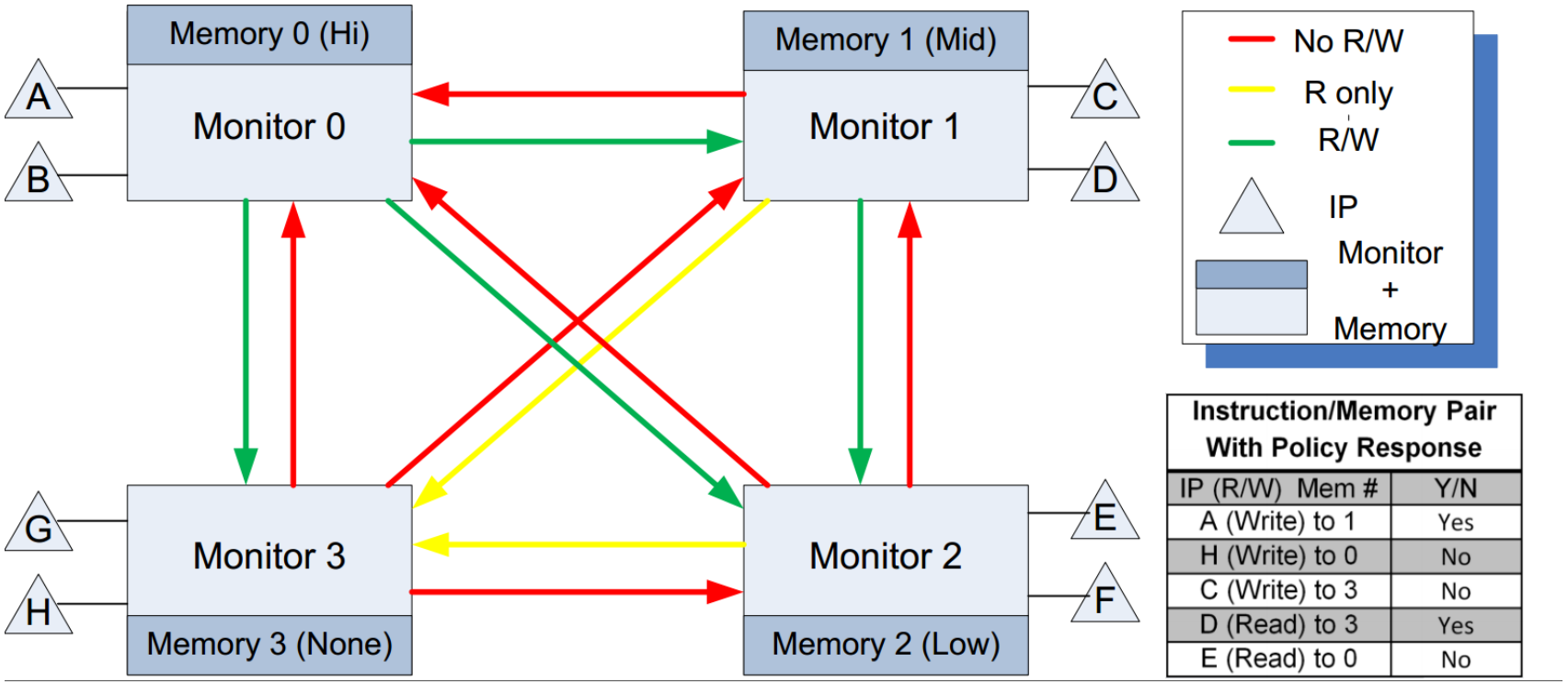
Security Policy

- Three components: MID, PID, A
  - Module ID: Identifier that specifies modules to access memory
  - Privilege ID: Designates privilege value based on trust
  - Action: Action to be performed on memory.
- Memory access can then be defined as
  - *MemAccess* = (*MID, A, Data, Addr*)
- This memory access is legal when, after a request, the MID is compared with a table of privileges and the comparison yields a true value.
- This security policy language allows us to be able to structure a design that disallows unauthorized access.

Separation Kernel

- Isolation technique that divides all resources under its control into blocks.
  - The actions of an active user in one block are isolated from another user in another block, unless an explicit means for that communication has been established.
- A separation kernel achieves isolation of different blocks by virtualizing shared resources.
  - To each user, each block appears to be completely accessible, but a security policy has ultimate control.
- This implemented separation kernel makes sure that:
  - Memory is allocated so users can access non-overlapping data
  - No simultaneous read-write access in memory by two users

Security Policy Implementing Kernel



Instruction/Memory Pair With Policy Response	
IP (R/W) Mem #	Y/N
A (Write) to 1	Yes
H (Write) to 0	No
C (Write) to 3	No
D (Read) to 3	Yes
E (Read) to 0	No

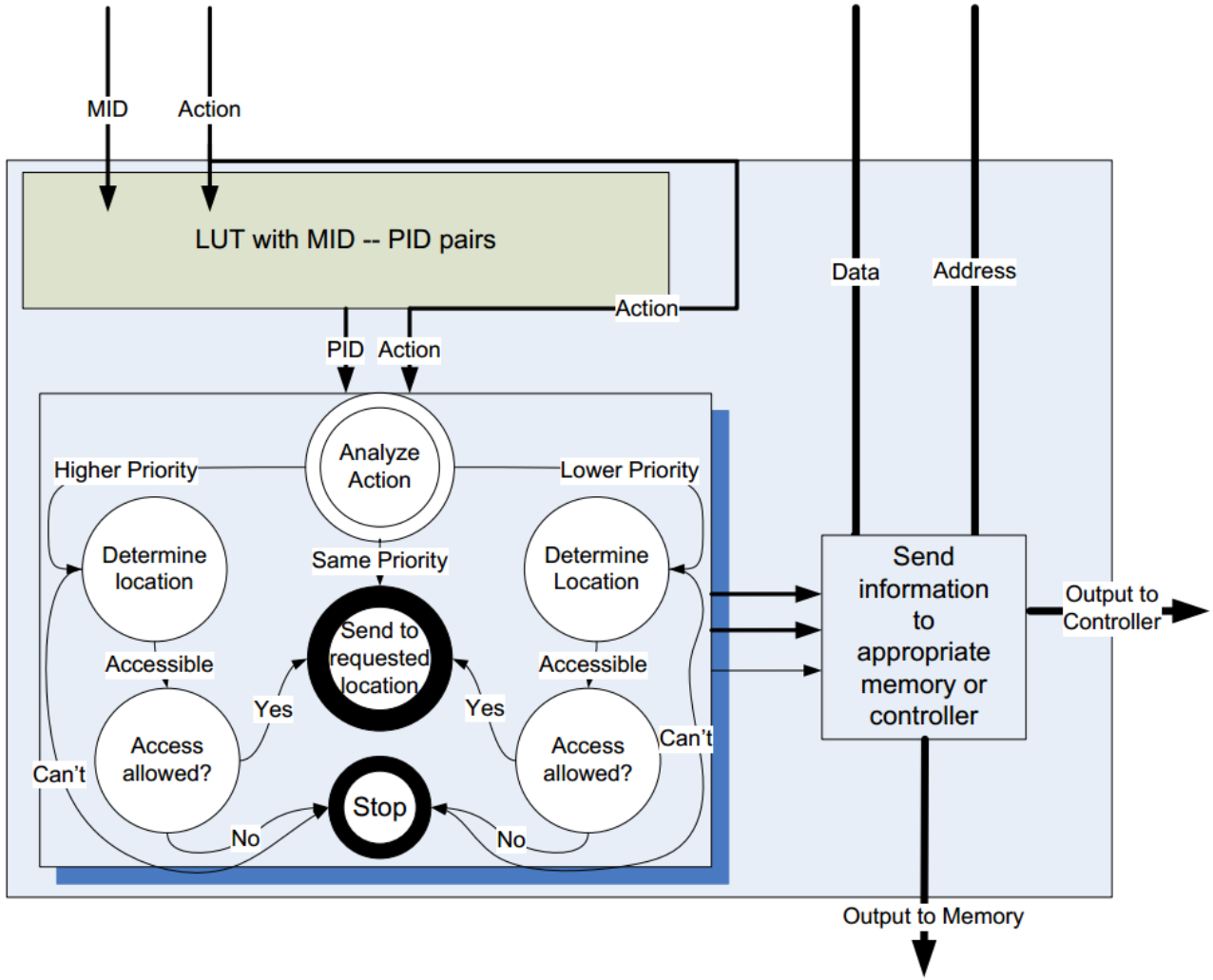
Advantages:

- It is fast!
  - A fast state machine can be made to satisfy requirements
- Can be implemented with very little resource overhead
  - No BlockRAM is necessary for this solution
- Provides protection from unwanted access
  - Shared memory schemes are particularly benefitted
- It can be easily implemented in an existing design.

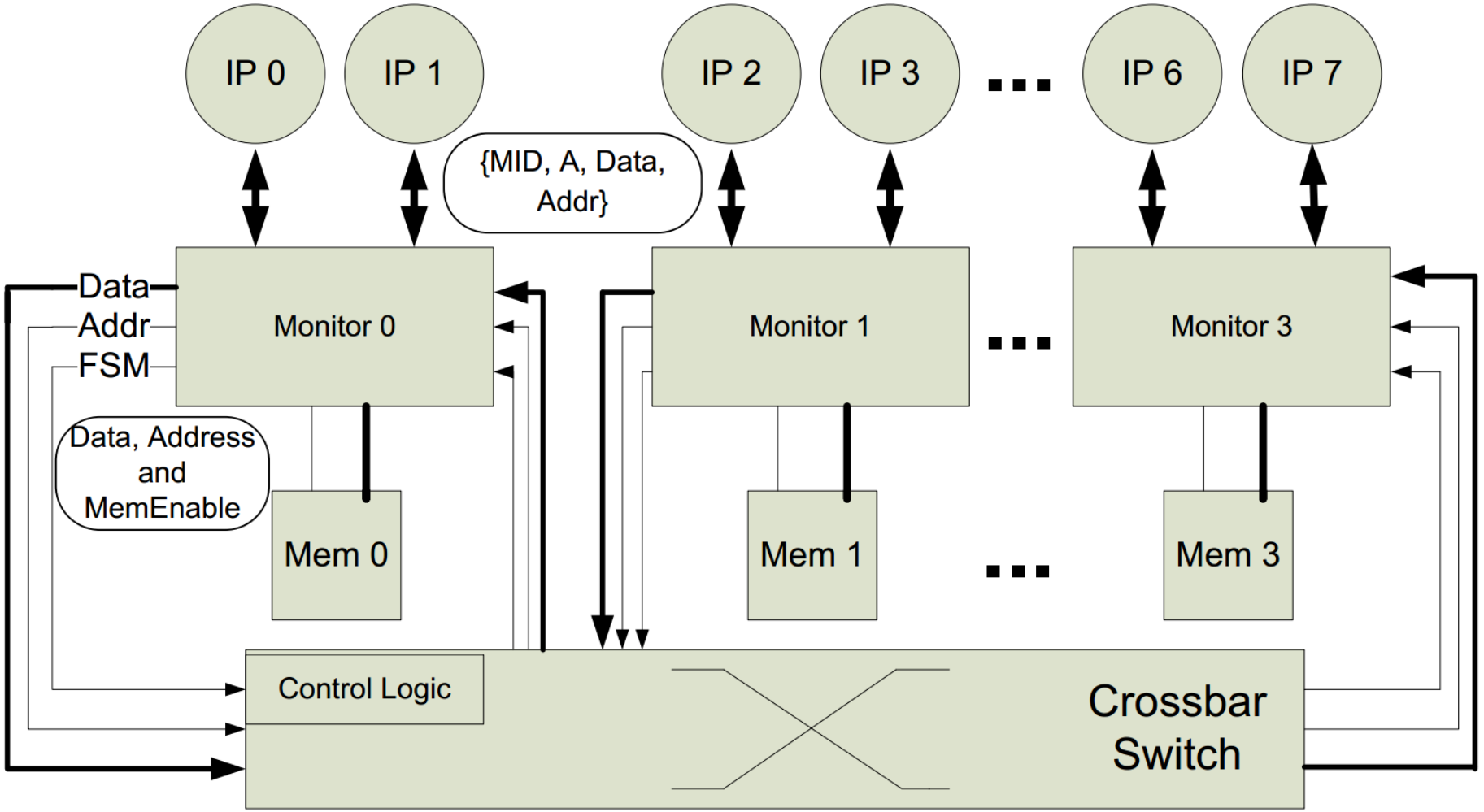
Reference Monitor

- Consists of a look-up table, a Finite-State Machine, and an arbiter
  - LUT has the PID for each MID.
  - Arbiter decides which IP goes in the FSM
  - FSM determines whether to allow or deny the memory access.
- Connects to two IPs each, and for communications between monitors, a crossbar switch was built.
- Each monitor connects to a Block RAM, creating a kernel block per each Block RAM
- Strict security concepts were implemented:
  - Internal components are isolated & data flushed out of buses

Reference Monitor



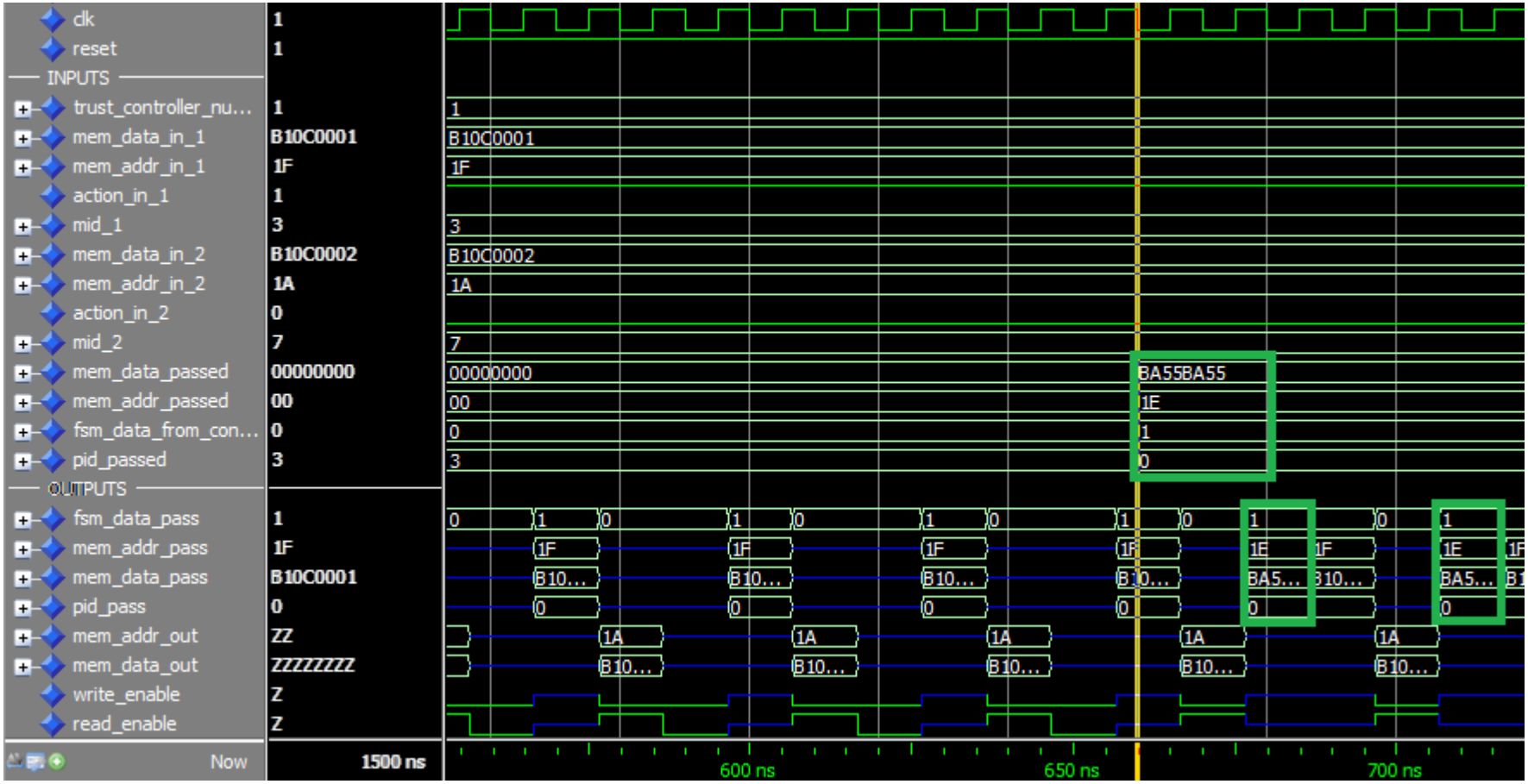
The Design


IP 0
IP 1
IP 2
IP 3
IP 6
IP 7
{MID, A, Data, Addr}
Data
Addr
FSM
Monitor 0
Monitor 1
Monitor 3
Data, Address and MemEnable
Mem 0
Mem 1
Mem 3
Control Logic
Crossbar Switch

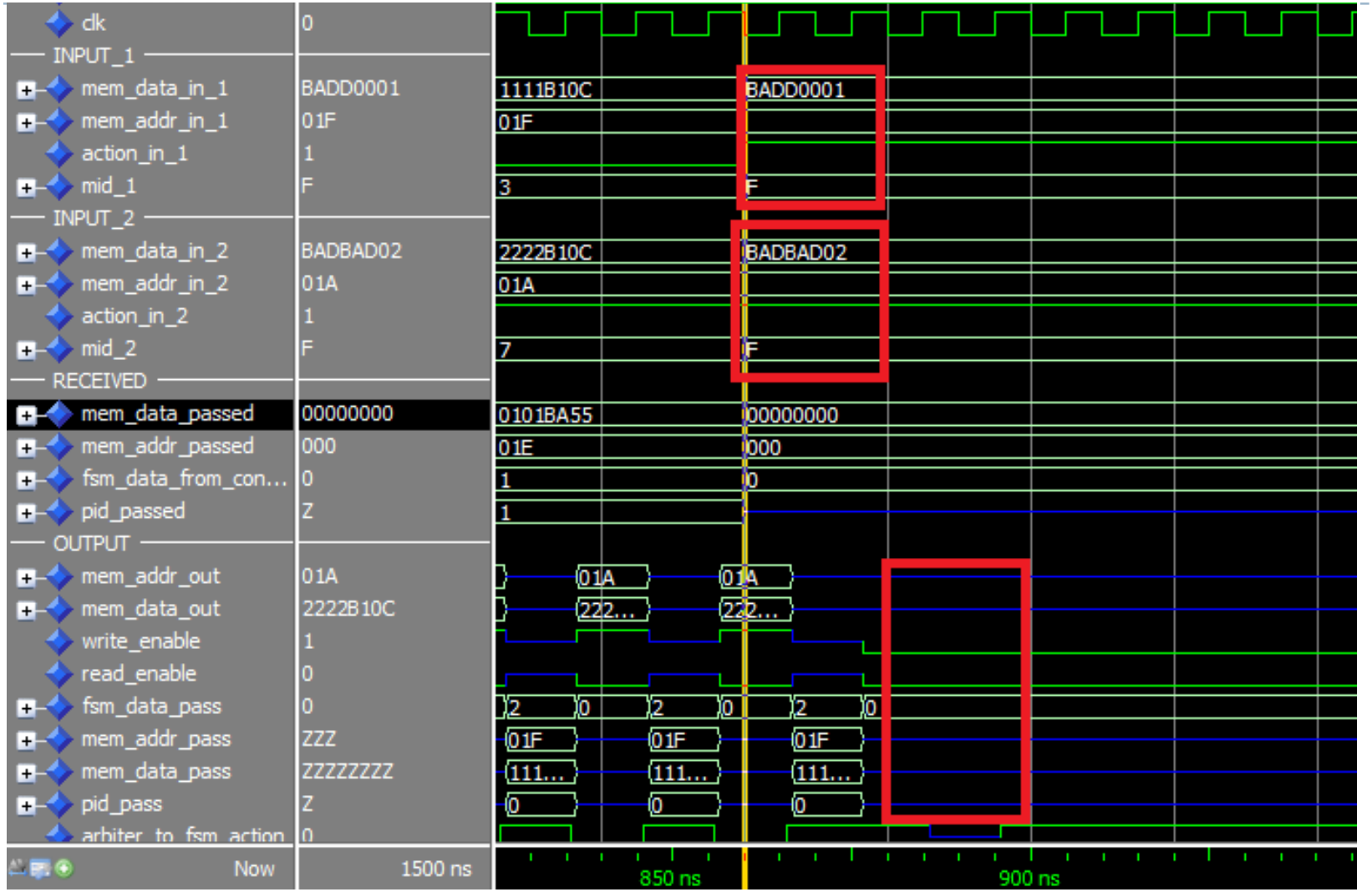
Implementation

- For the implementation of our separation kernel, the on-chip memory block illustrated before was used as a true dual-port RAM memory
- The architecture was developed on a Xilinx Virtex-6 XC6VLX240T-1FF116 board, by using the Xilinx Design Suite ISE 13.4.
- This entire system was tested at a operative frequency of 100 MHz using ModelSim SE 6.6f
- Its effective frequency in the physical device was 500 MHz

Results


clk
reset
INPUTS
trust_controller_nu...
mem_data_in_1
mem_addr_in_1
action_in_1
mid_1
mem_data_in_2
mem_addr_in_2
action_in_2
mid_2
mem_data_passed
mem_addr_passed
fsm_data_from_con...
pid_passed
OUTPUTS
fsm_data_pass
mem_addr_pass
mem_data_pass
pid_pass
mem_addr_out
mem_data_out
write_enable
read_enable
Now
1500 ns
600 ns
650 ns
700 ns
BA55BA55

Results


clk
0
INPUT_1
mem_data_in_1
BADD0001
mem_addr_in_1
01F
action_in_1
1
mid_1
F
INPUT_2
mem_data_in_2
BADBAD02
mem_addr_in_2
01A
action_in_2
1
mid_2
F
RECEIVED
mem_data_passed
00000000
mem_addr_passed
000
fsm_data_from_con...
0
pid_passed
Z
OUTPUT
mem_addr_out
01A
mem_data_out
2222B10C
write_enable
1
read_enable
0
fsm_data_pass
0
mem_addr_pass
ZZZ
mem_data_pass
ZZZZZZZZ
pid_pass
Z
arbiter_to_fsm_action
0
Now
1500 ns
1111B10C
BADD0001
01F
3
F
2222B10C
BADBAD02
01A
7
F
0101BA55
00000000
01E
000
1
0
1
01A
01A
222...
222...
2
0
2
0
2
0
01F
01F
01F
111...
111...
111...
0
0
0
850 ns
900 ns

Analysis

- The separation kernel designed has only one transition state between memory access.
- The number of clock cycles inside the monitor itself has been reduced by implementing pipelining between the internal components.
- This allows for the monitor to be used in throughput-heavy and low-latency applications, such as burst-read or burst-write memory access.

Resource	Entire Design		Monitor Only	
	Used/Available	% Used	Used/Available	% Used
Slice Registers	1,332/301,440	1%	360/301,440	1%
LUTs	11,601/150,720	7%	588/150,720	1%
Block RAMs	8, 192/58,400	14%	0/58,400	0%

Performance On	Original	Worked Design
Delay	1 Clock cycle	3 Clock cycles (average)
Size Overhead	None	Very Small
Security Achieved	None	Separation Kernel Design

Conclusion

- The work provided here shows a memory security scheme that has been designed and implemented for on-chip memory inside FPGAs.
- The simulations show that the separation kernel that was designed is successful in securing the on-chip memory from unauthorized accesses from IPs whom are untrusted.
- The work here can be expanded to include other security concerns such as integrity checks for the memory to detect tampered memory values, and the implementation of a different architecture to organize the communication between monitors.

References

- [1] S. Mal-Sarkar and A. Krishna, "Hardware trojan attacks in FPGA devices: threat analysis and effective counter measures," in Proc. 24th edition of the great lakes symposium on VLSI (GLSVLSI '14), 2014, pp. 287–292.
- [2] L. Fiorin and S. Lukovic, "Implementation of a reconfigurable data protection module for NoC-based MPSoCs," in IEEE Int'l Symp. On Parallel and Distributed Processing (IPDPS '08), 2008, pp. 14–18.
- [3] T. Levin and C. Irvine, "A Least Privilege Model for Static Separation Kernels," in E-business and Telecommunication Networks, vol. 9 of Communications in Computer and Information Science, 2008, pp. 146–158.
- [4] T. Huffmire and S. Prasad, "Policy-Driven Memory Protection for Reconfigurable Hardware," in Proc. of 11th European Symposium on Research in Computer Security, 2008, pp. 461–478.
- [5] XILINX. (2011) Virtex-6 FPGA Memory Resources User Guide. ug363.pdf. [Online]. Available: http://www.xilinx.com/support/documentation/user guides/
- [6] XILINX. (2015) UltraScale FPGA Overview. ds890.pdf [Online]. Available: http://www.xilinx.com/support/documentation/data sheets/ds890-ultrascale-overview.pdf