BIST for Arm’s 3nm Multi-Port Register Files

Andy Chen   Vianney Choserot
Munish Kumar Khushal Gelda
Shreyas Pramod Dixit

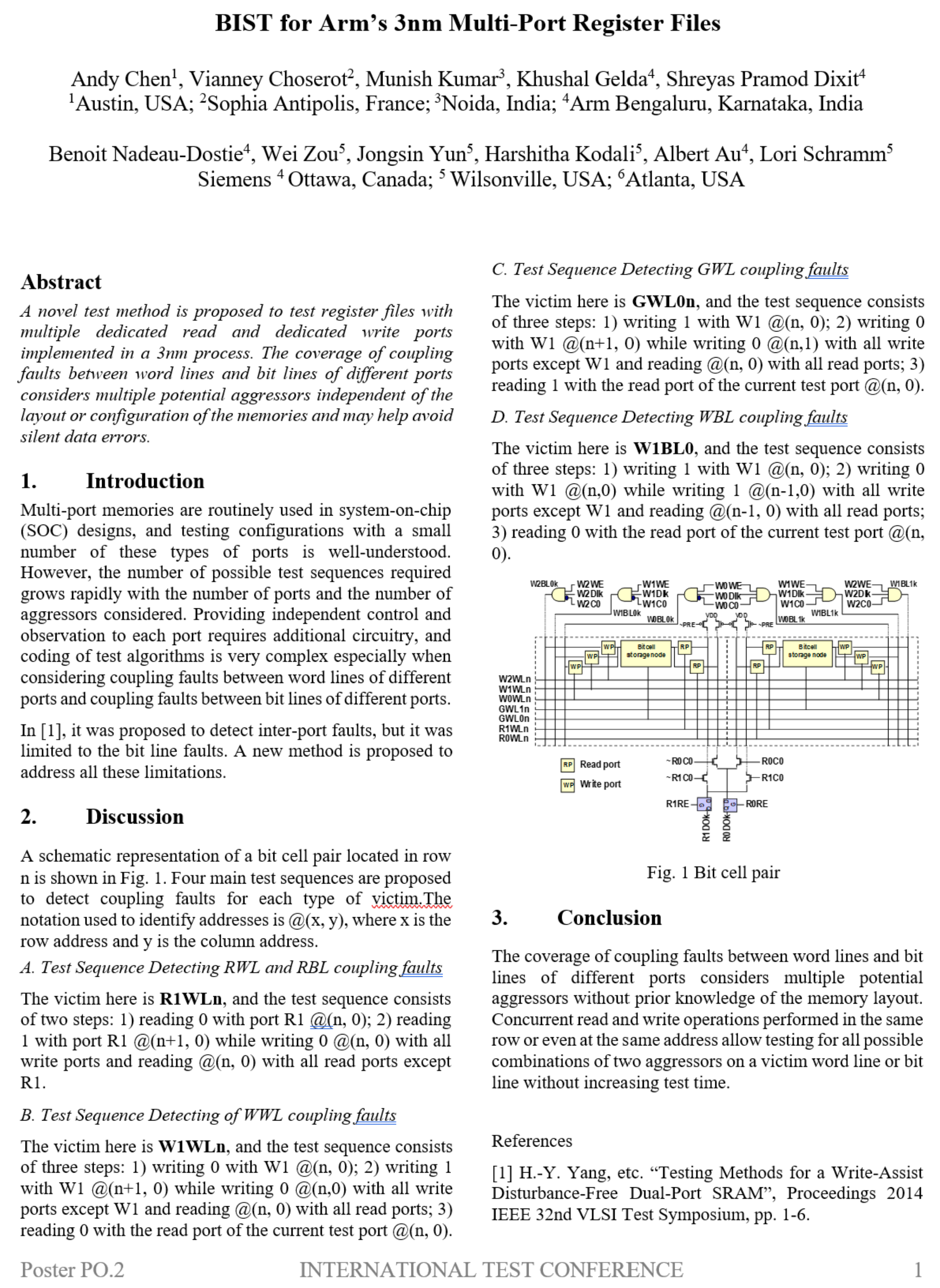
Benoit Nadeau-Dostie   Wei Zou
Jongsin Yun Harshitha Kodali
Albert Au   Lori Schramm

arm

SIEMENS

Outline

- General architecture of muti-port register files (MPRF)
- MPRF testing challenges
- Our proposed method
- Conclusion



BIST for Arm's 3nm Multi-Port Register Files

Andy Chen[1], Vianney Choserot[2], Munish Kumar[3], Khushal Gelda[4], Shreyas Pramod Dixit[4]
[1]Austin, USA; [2]Sophia Antipolis, France; [3]Noida, India; [4]Arm Bengaluru, Karnataka, India

Benoit Nadeau-Dostie[4], Wei Zou[5], Jongsin Yun[5], Harshitha Kodali[5], Albert Au[4], Lori Schramm[5]
Siemens [4] Ottawa, Canada; [5] Wilsonville, USA; [6]Atlanta, USA

Abstract

A novel test method is proposed to test register files with multiple dedicated read and dedicated write ports implemented in a 3nm process. The coverage of coupling faults between word lines and bit lines of different ports considers multiple potential aggressors independent of the layout or configuration of the memories and may help avoid silent data errors.

1. Introduction

Multi-port memories are routinely used in system-on-chip (SOC) designs, and testing configurations with a small number of these types of ports is well-understood. However, the number of possible test sequences required grows rapidly with the number of ports and the number of aggressors considered. Providing independent control and observation to each port requires additional circuitry, and coding of test algorithms is very complex especially when considering coupling faults between word lines of different ports and coupling faults between bit lines of different ports.

In [1], it was proposed to detect inter-port faults, but it was limited to the bit line faults. A new method is proposed to address all these limitations.

2. Discussion

A schematic representation of a bit cell pair located in row n is shown in Fig. 1. Four main test sequences are proposed to detect coupling faults for each type of victim.The notation used to identify addresses is @(x, y), where x is the row address and y is the column address.

A. Test Sequence Detecting RWL and RBL coupling faults

The victim here is **R1WLn**, and the test sequence consists of two steps: 1) reading 0 with port R1 @(n, 0); 2) reading 1 with port R1 @(n+1, 0) while writing 0 @(n, 0) with all write ports and reading @(n, 0) with all read ports except R1.

B. Test Sequence Detecting of WWL coupling faults

The victim here is **W1WLn**, and the test sequence consists of three steps: 1) writing 0 with W1 @(n, 0); 2) writing 1 with W1 @(n+1, 0) while writing 0 @(n,0) with all write ports except W1 and reading @(n, 0) with all read ports; 3) reading 0 with the read port of the current test port @(n, 0).

C. Test Sequence Detecting GWL coupling faults

The victim here is **GWL0n**, and the test sequence consists of three steps: 1) writing 1 with W1 @(n, 0); 2) writing 0 with W1 @(n+1, 0) while writing 0 @(n,1) with all write ports except W1 and reading @(n, 0) with all read ports; 3) reading 1 with the read port of the current test port @(n, 0).

D. Test Sequence Detecting WBL coupling faults

The victim here is **W1BL0**, and the test sequence consists of three steps: 1) writing 1 with W1 @(n, 0); 2) writing 0 with W1 @(n,0) while writing 1 @(n-1,0) with all write ports except W1 and reading @(n-1, 0) with all read ports; 3) reading 0 with the read port of the current test port @(n, 0).

Fig. 1 Bit cell pair

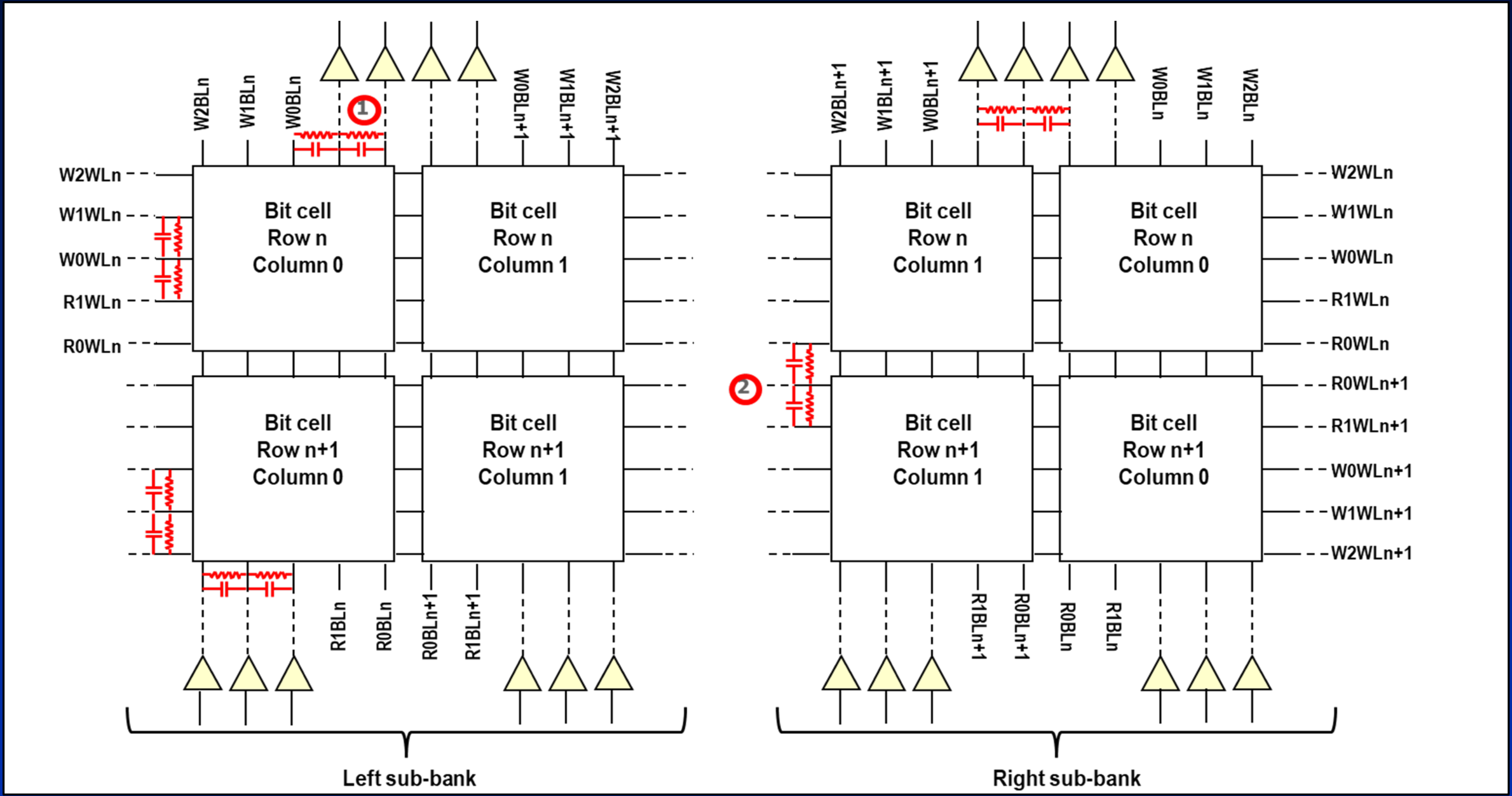
3. Conclusion

The coverage of coupling faults between word lines and bit lines of different ports considers multiple potential aggressors without prior knowledge of the memory layout. Concurrent read and write operations performed in the same row or even at the same address allow testing for all possible combinations of two aggressors on a victim word line or bit line without increasing test time.

References

[1] H.-Y. Yang, etc. "Testing Methods for a Write-Assist Disturbance-Free Dual-Port SRAM", Proceedings 2014 IEEE 32nd VLSI Test Symposium, pp. 1-6.

General architecture of MPRF



- Testing complexity grows with increasing number of ports

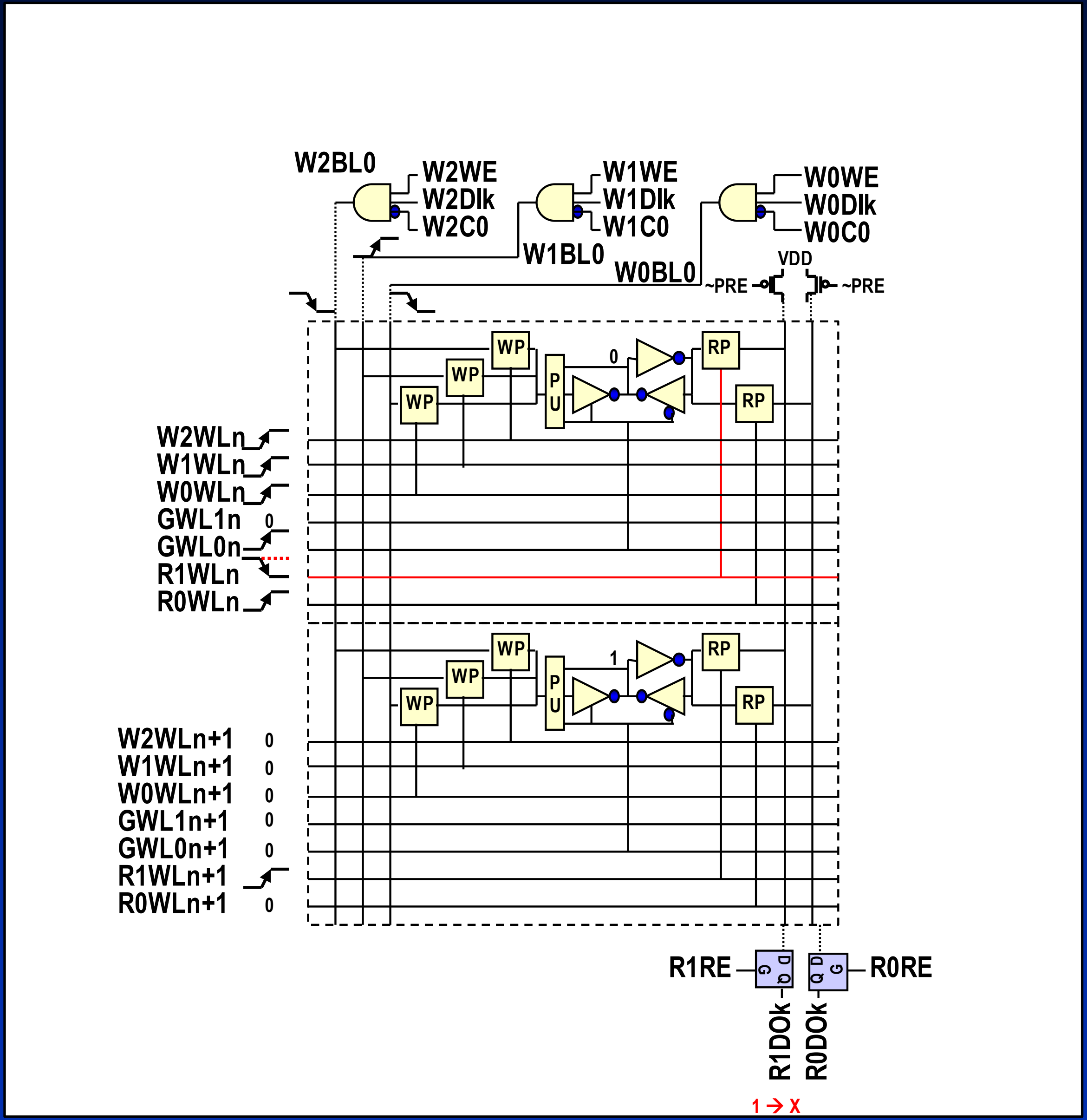
MPRF testing challenges

- The **number of ports** is large
- Providing **independent control** and **observation** to each port requires a lot of circuitry
- Coding **test algorithms** is very **complex**, especially considering **coupling faults** between word lines of different ports and coupling faults between bit lines of different ports
- The number of possible **test sequences** grows rapidly with the number of ports and the **number of aggressors** considered, especially if the memory layout is not known

The proposed method

- All the ports share the same address generator and data generator to **reduce the hardware overhead**
- Performing **concurrent write** and **read operations** at the same row, or even the same address
  - can **detect the coupling faults** with one and two aggressors
  - will **reduce** the **test time** without enumerating all possible coupling fault combinations
- **No layout information** is needed

Detection of RWL coupling faults

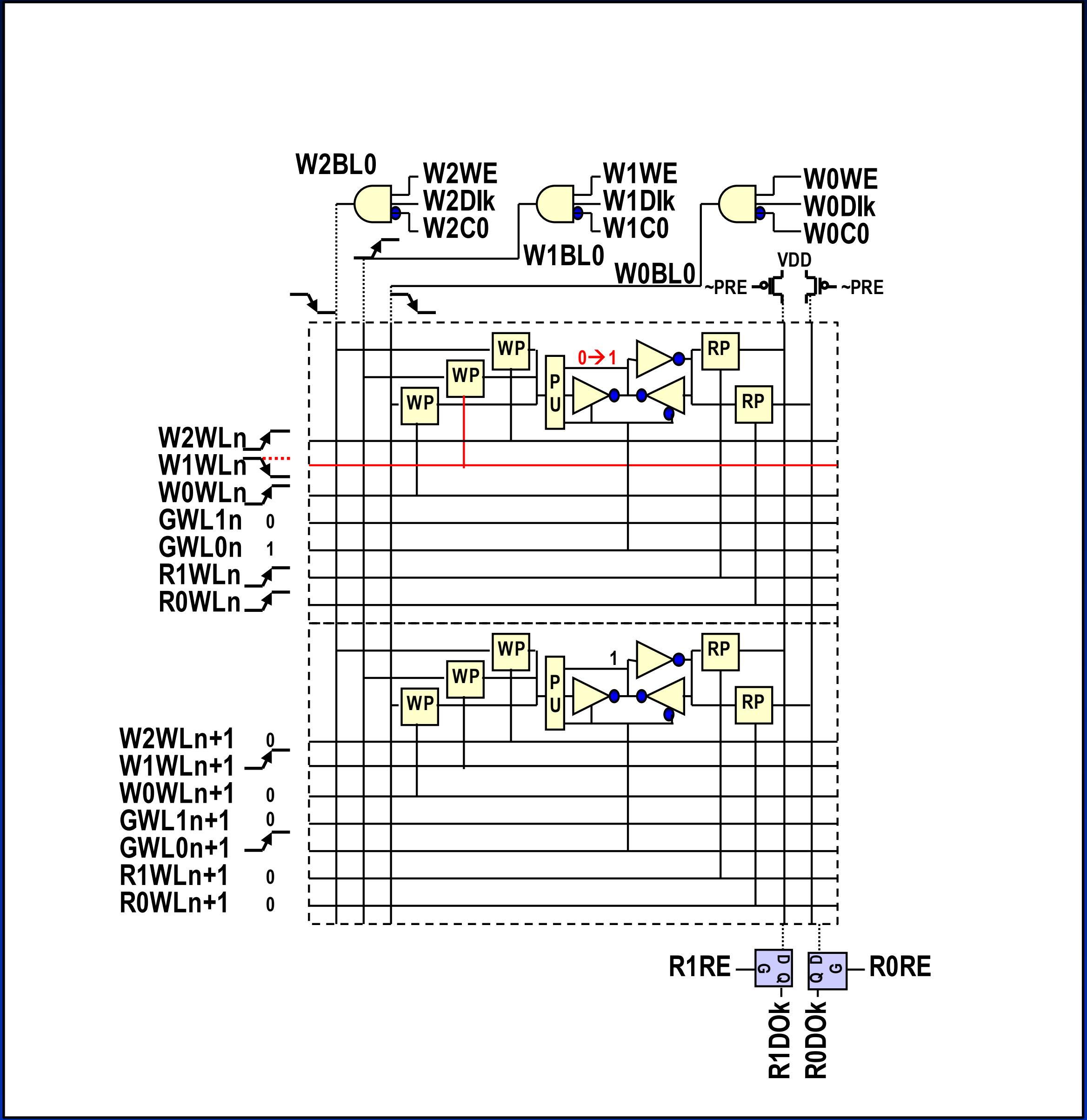


- **Victim: R1WLn**

1) reading 0 with port R1 @(n, 0)

2) reading 1 with port R1 @(n+1, 0) while writing 0 @(n, 0) with all write ports and reading @(n, 0) with all read ports except R1

Detection of WWL coupling faults



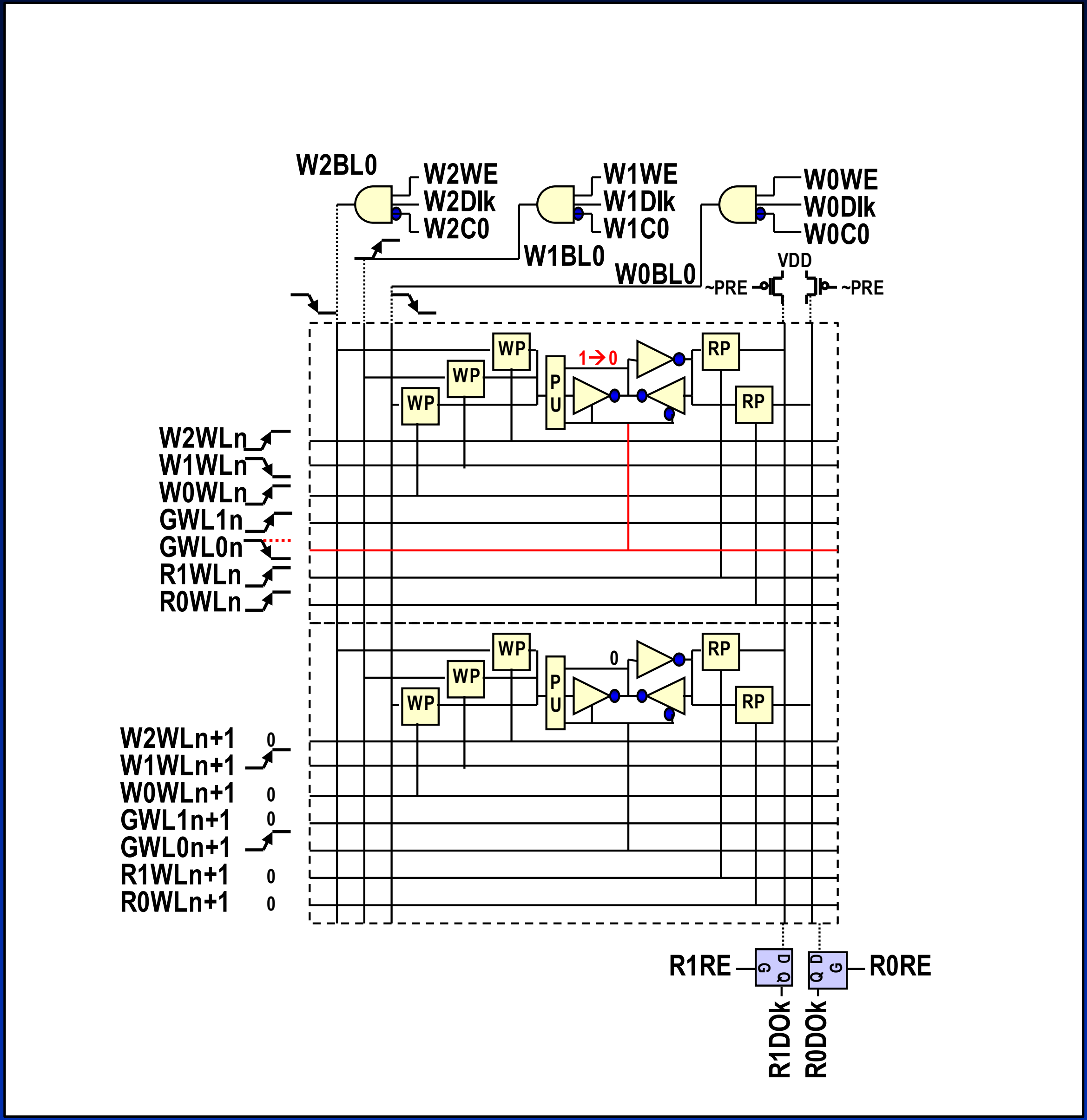
- **Victim: W1WLn**

1) writing 0 with W1 @(n, 0)

2) writing 1 with W1 @(n+1, 0) while writing 0 @(n,0) with all write ports except W1 and reading @(n, 0) with all read ports

3) reading 0 with the read port of the current test port @(n, 0)

Detection of GWL coupling faults



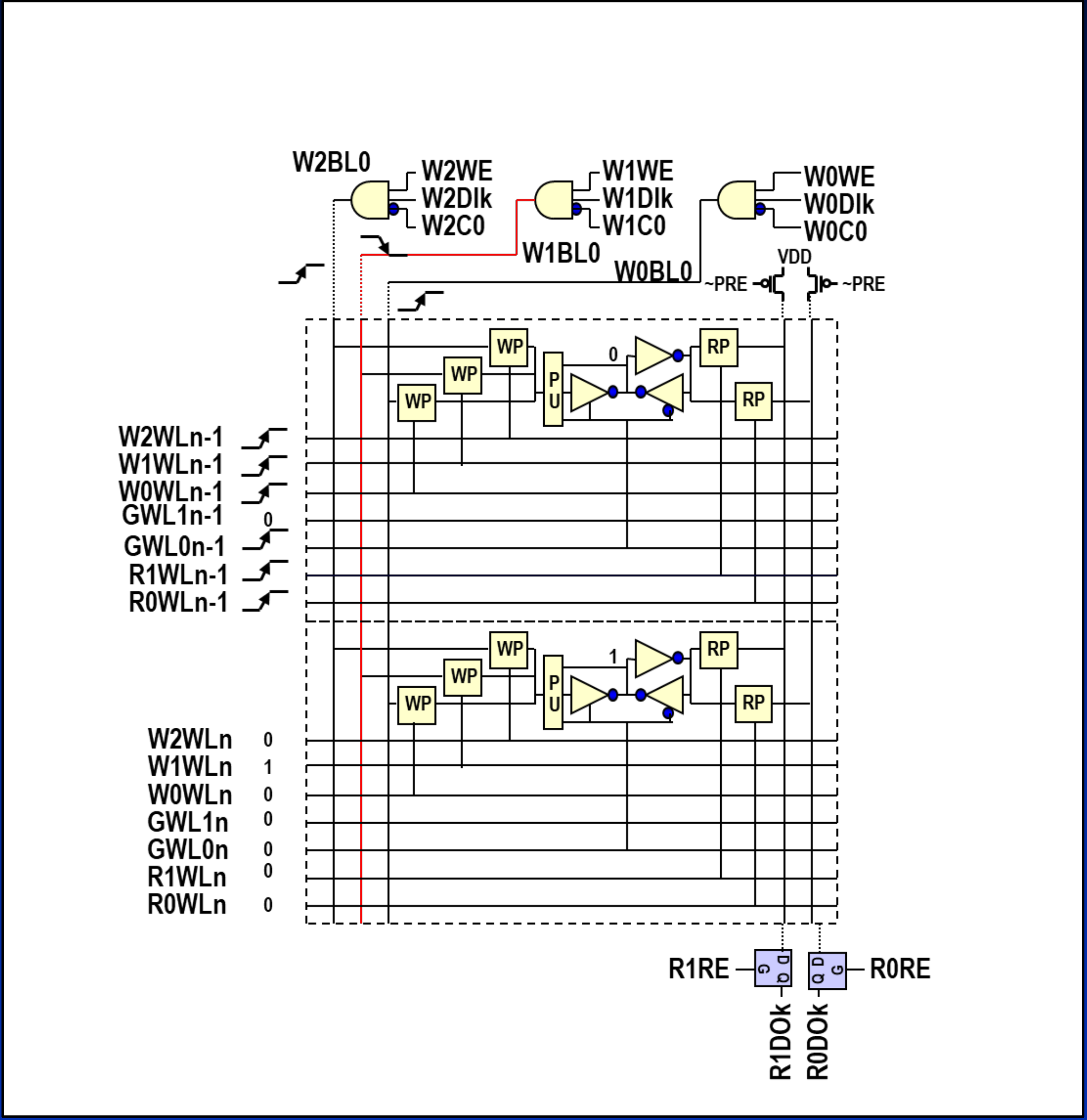
- Victim: GWL0n

1) writing 1 with W1 @(n, 0)

2) writing 0 with W1 @(n+1, 0) while writing 0 @(n,1) with all write ports except W1 and reading @(n, 0) with all read ports

3) reading 1 with the read port of the current test port @(n, 0)

Detection of WBL coupling faults



- **Victim: W1BL0**

1) writing 1 with W1 @(n, 0)

2) writing 0 with W1 @(n,0) while writing 1 @(n-1,0) with all write ports except W1 and reading @(n-1, 0) with all read ports

3) reading 0 with the read port of the current test port @(n, 0)

Detection of RBL coupling faults

- The process is the same as the one used to detect RWL coupling faults
- The bit lines of all ports performing concurrent operations are driven to a value which is the inverse of the read port under test
- The address sequence causes transitions on all bit lines to sensitize potential coupling faults

Conclusion

Concurrent read and **write** operations performed in the same row – or even at the same address – allow testing for **all possible** combinations of two **aggressors** on a victim word line or bit line **without increasing test time**