An Architecture for Combined Test Data Compression and Abort-on-Fail Test

Erik Larsson and Jon Persson

Embedded Systems Laboratory
Department of Computer Science
Linköping University
Sweden

Purpose

Problems when testing ICs:

- Long test times
- High ATE memory requirement
- Low throughput

+ **Multi-site testing** increases throughput

- Requires ATE memory

+ **Abort-on-fail testing** lower testing times

- Test data volume remains large

+ **Test data compression** lower ATE memory

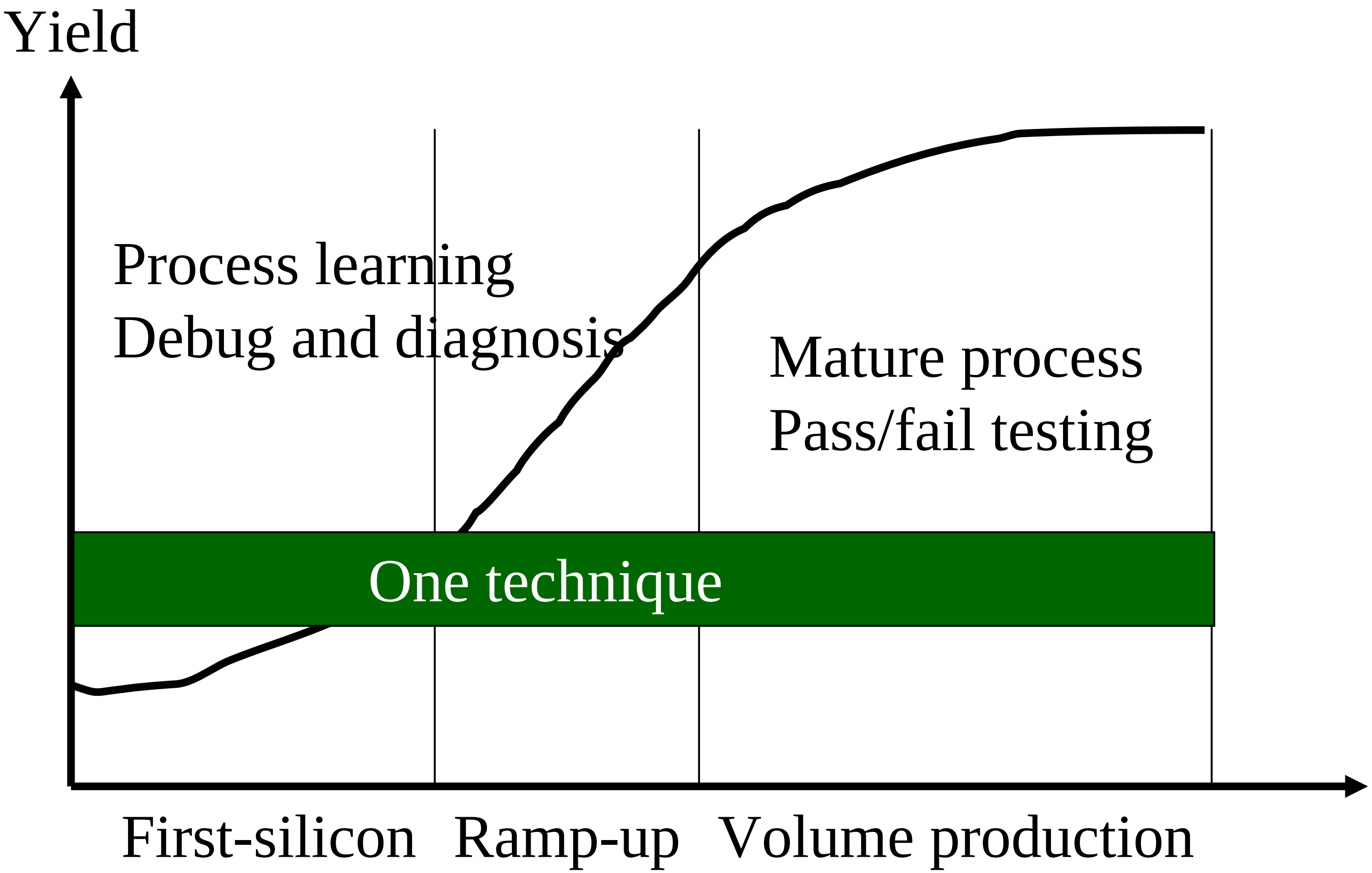
- MISRs cannot terminate immediately when a fault is detected

- Unknowns (X) must be handled

Aim: Define an architecture that allows:

1. High degree of multi-site testing
2. Test data volume compression, and
3. Abort-on-Fail testing

Purpose



Yield

Process learning
Debug and diagnosis

Mature process
Pass/fail testing

One technique

First-silicon Ramp-up Volume production

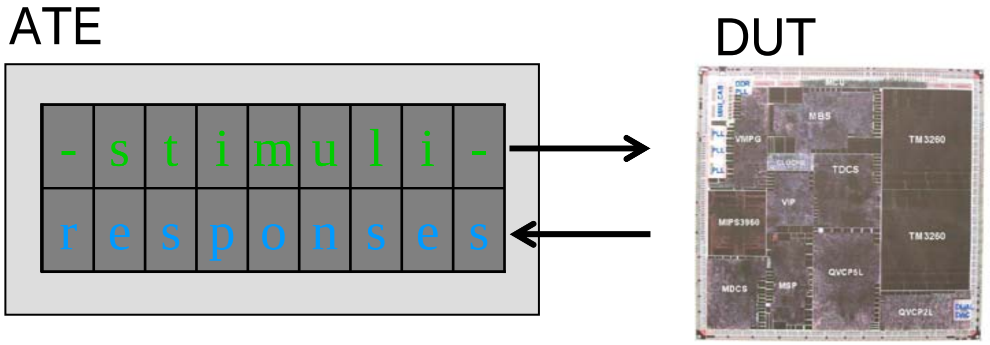
Outline

1. Introduction
2. Prior Work
3. Test Architecture
4. Experimental Results
5. Conclusion

1. Introduction

Testing

- Given:
  - DUT (Device-under-Test)
  - ATE (Automatic Test Equipment)
  - Test stimuli
  - Test responses



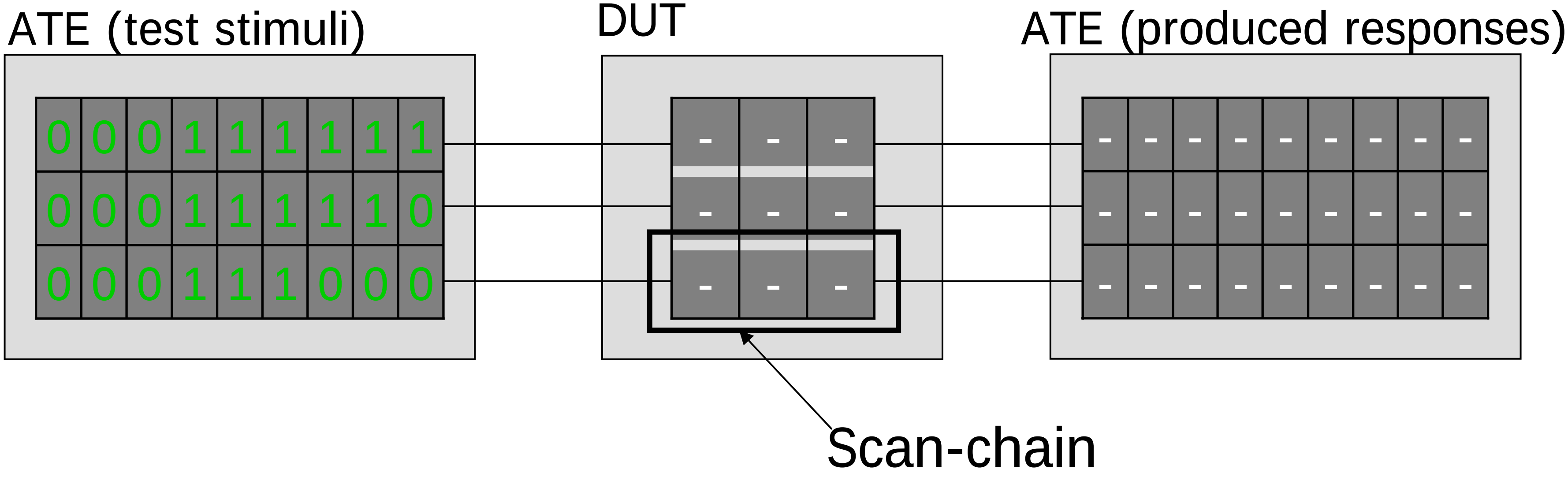
Testing

vector {stimuli} {expected responses}

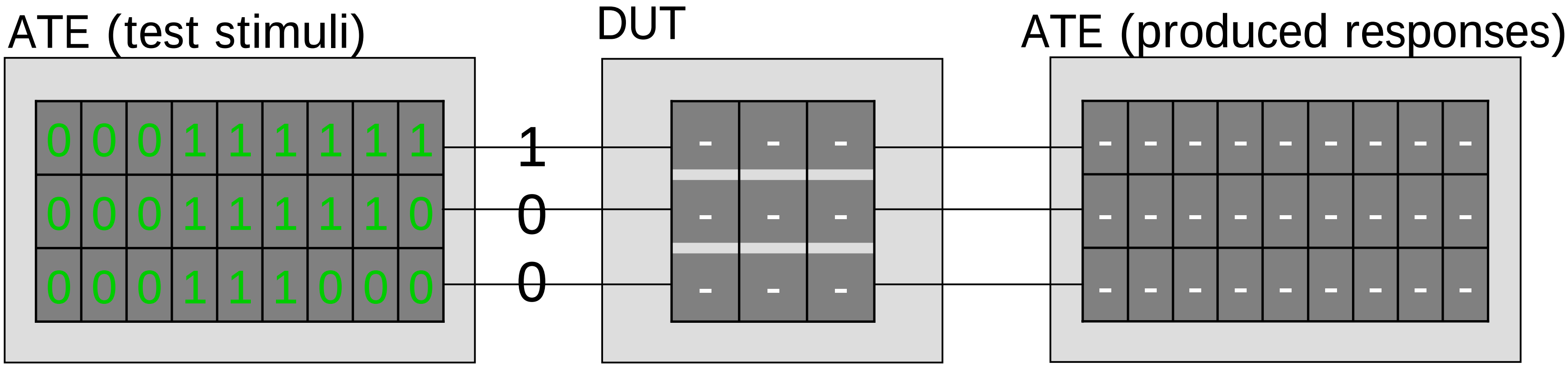
1 {111 110 000} {000 000 011}

2 {111 111 111} {111 000 000}

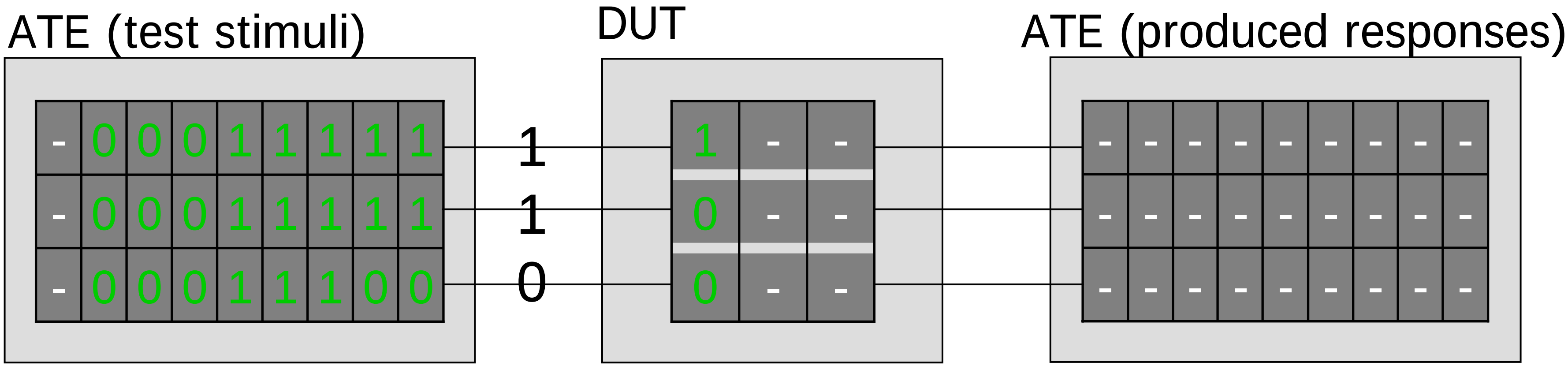
3 {000 000 000} {000 111 111}



Testing


ATE (test stimuli)
DUT
ATE (produced responses)
0 0 0 1 1 1 1 1 1
0 0 0 1 1 1 1 1 0
0 0 0 1 1 1 0 0 0
1
0
0
- - -
- - -
- - -
- - - - - - - - -
- - - - - - - - -
- - - - - - - - -

Testing



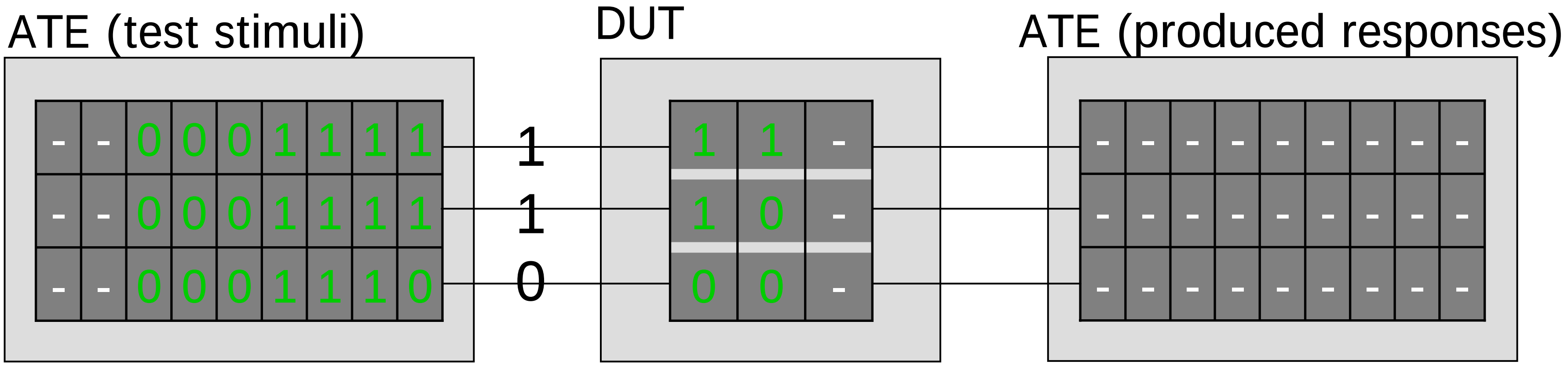
-	0	0	0	1	1	1	1	1
-	0	0	0	1	1	1	1	1
-	0	0	0	1	1	1	0	0

1	-	-
0	-	-
0	-	-

-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

1. Introduction

Testing



ATE (test stimuli)

-	-	0	0	0	1	1	1	1
-	-	0	0	0	1	1	1	1
-	-	0	0	0	1	1	1	0

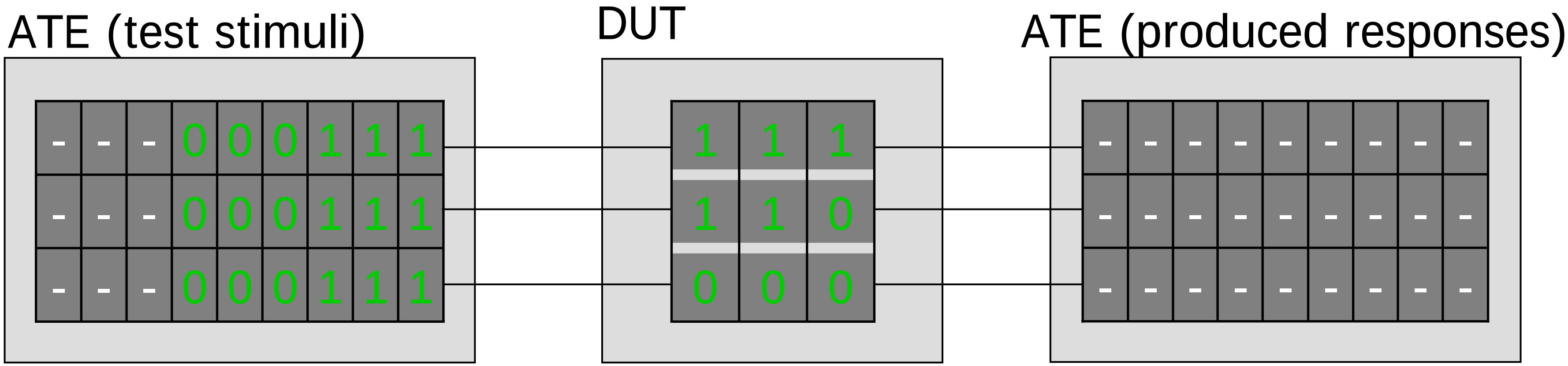
DUT

1	1	-
1	0	-
0	0	-

ATE (produced responses)

-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

Testing



ATE (test stimuli)

- - - 0 0 0 1 1 1
- - - 0 0 0 1 1 1
- - - 0 0 0 1 1 1

DUT

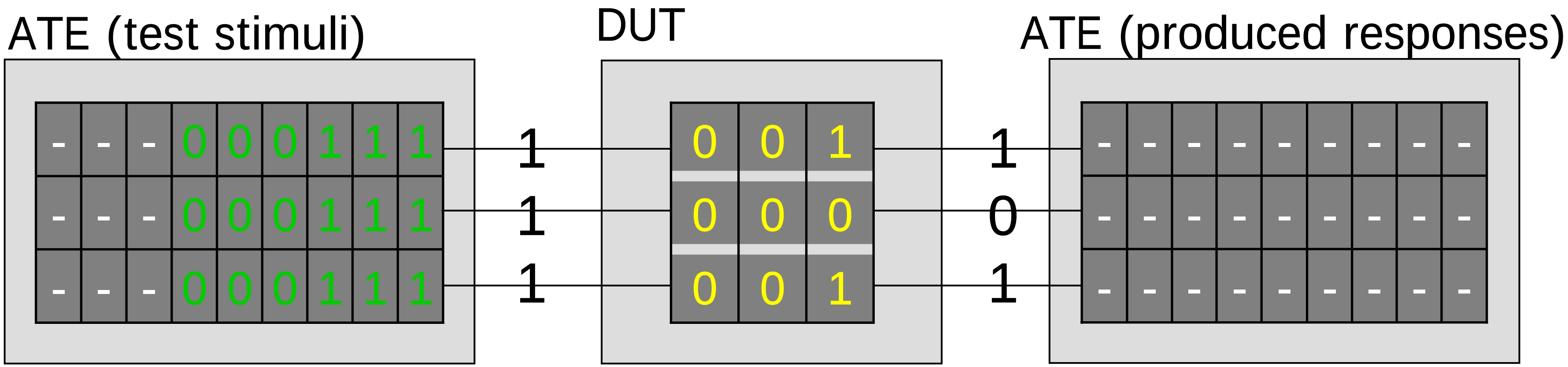
1 1 1
1 1 0
0 0 0

ATE (produced responses)

- - - - - - - - -
- - - - - - - - -
- - - - - - - - -

1. Introduction

Testing



ATE (test stimuli)

-	-	-	0	0	0	1	1	1
-	-	-	0	0	0	1	1	1
-	-	-	0	0	0	1	1	1

1
1
1

DUT

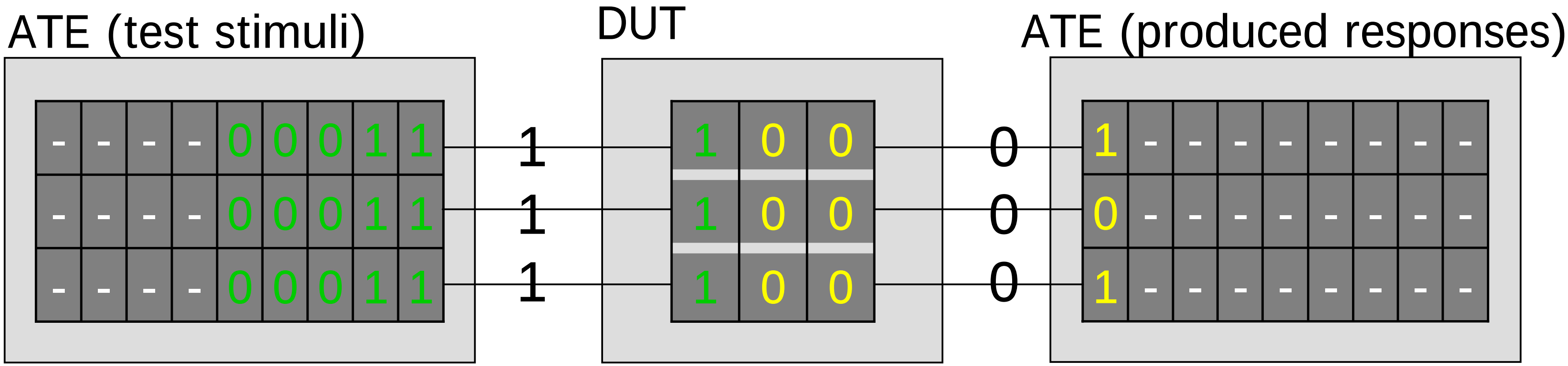
0	0	1
0	0	0
0	0	1

1
0
1

ATE (produced responses)

-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

Testing



ATE (test stimuli)

-	-	-	-	0	0	0	1	1
-	-	-	-	0	0	0	1	1
-	-	-	-	0	0	0	1	1

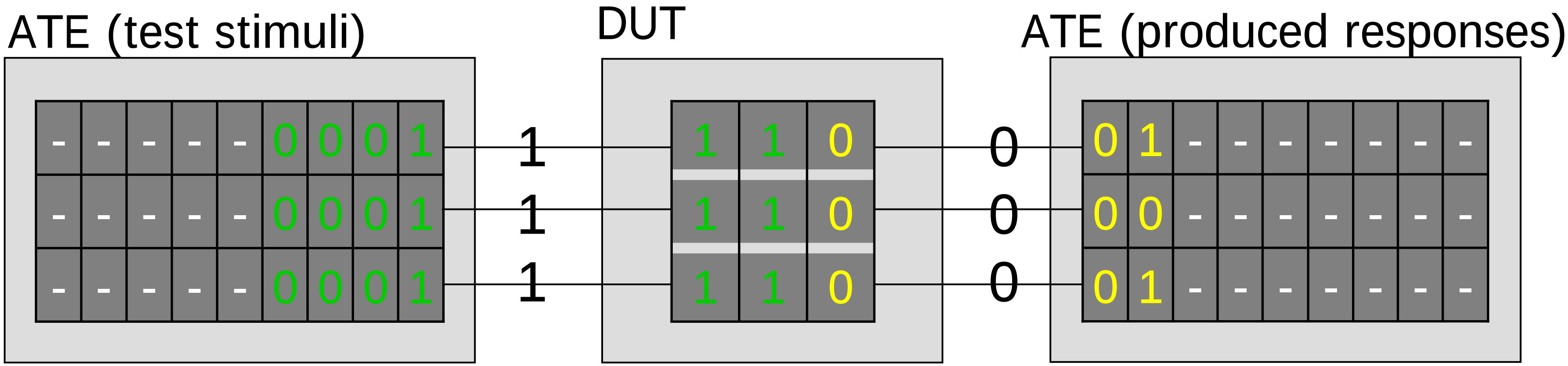
DUT

1	0	0
1	0	0
1	0	0

ATE (produced responses)

1	-	-	-	-	-	-	-	-
0	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-

Testing



ATE (test stimuli)

-	-	-	-	-	0	0	0	1
-	-	-	-	-	0	0	0	1
-	-	-	-	-	0	0	0	1

DUT

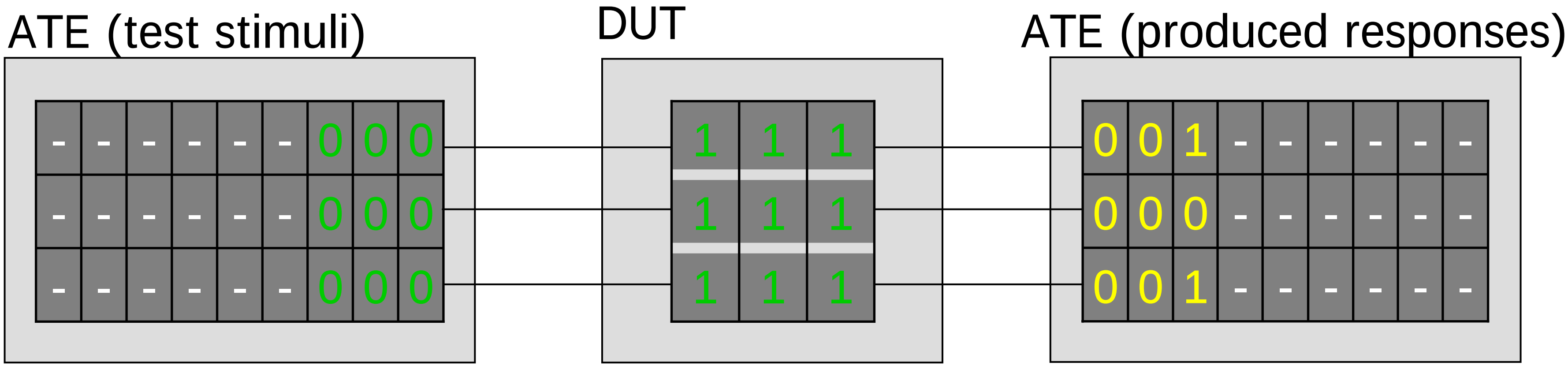
1	1	0
1	1	0
1	1	0

ATE (produced responses)

0	1	-	-	-	-	-	-	-
0	0	-	-	-	-	-	-	-
0	1	-	-	-	-	-	-	-

1. Introduction

Testing



ATE (test stimuli)

-	-	-	-	-	-	0	0	0
-	-	-	-	-	-	0	0	0
-	-	-	-	-	-	0	0	0

DUT

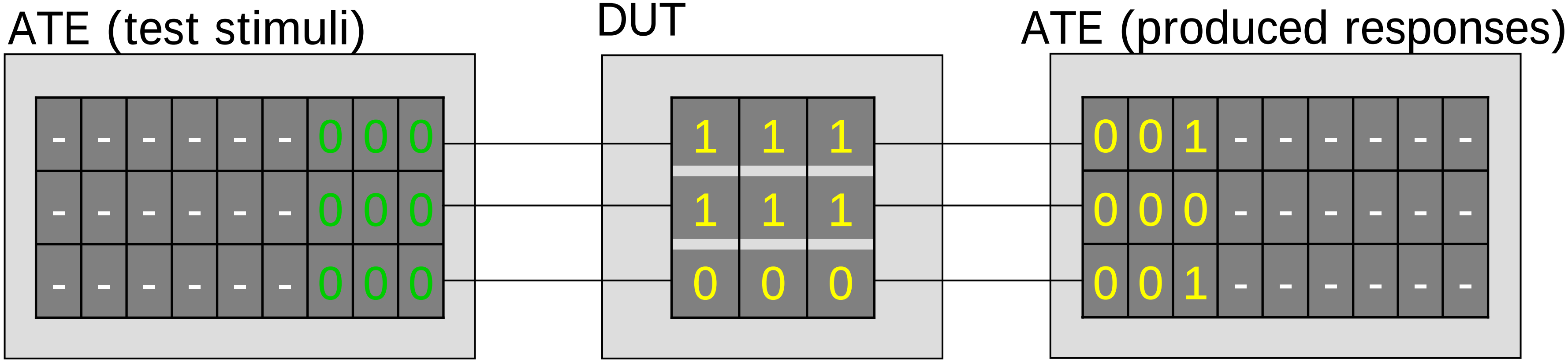
1	1	1
1	1	1
1	1	1

ATE (produced responses)

0	0	1	-	-	-	-	-	-
0	0	0	-	-	-	-	-	-
0	0	1	-	-	-	-	-	-

1. Introduction

Testing



ATE (test stimuli)

-	-	-	-	-	-	0	0	0
-	-	-	-	-	-	0	0	0
-	-	-	-	-	-	0	0	0

DUT

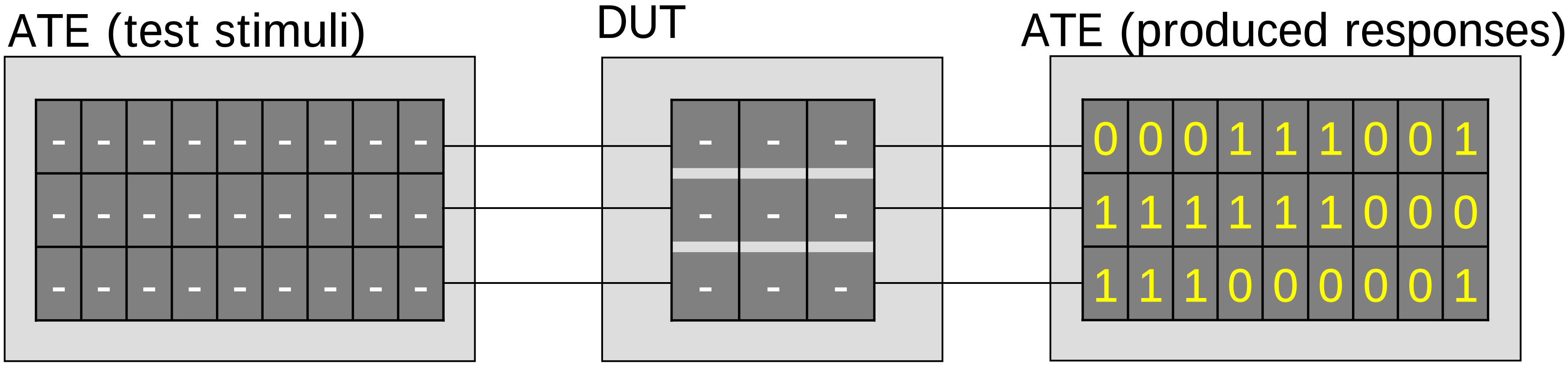
1	1	1
1	1	1
0	0	0

ATE (produced responses)

0	0	1	-	-	-	-	-	-
0	0	0	-	-	-	-	-	-
0	0	1	-	-	-	-	-	-

1. Introduction

Testing



ATE (test stimuli)

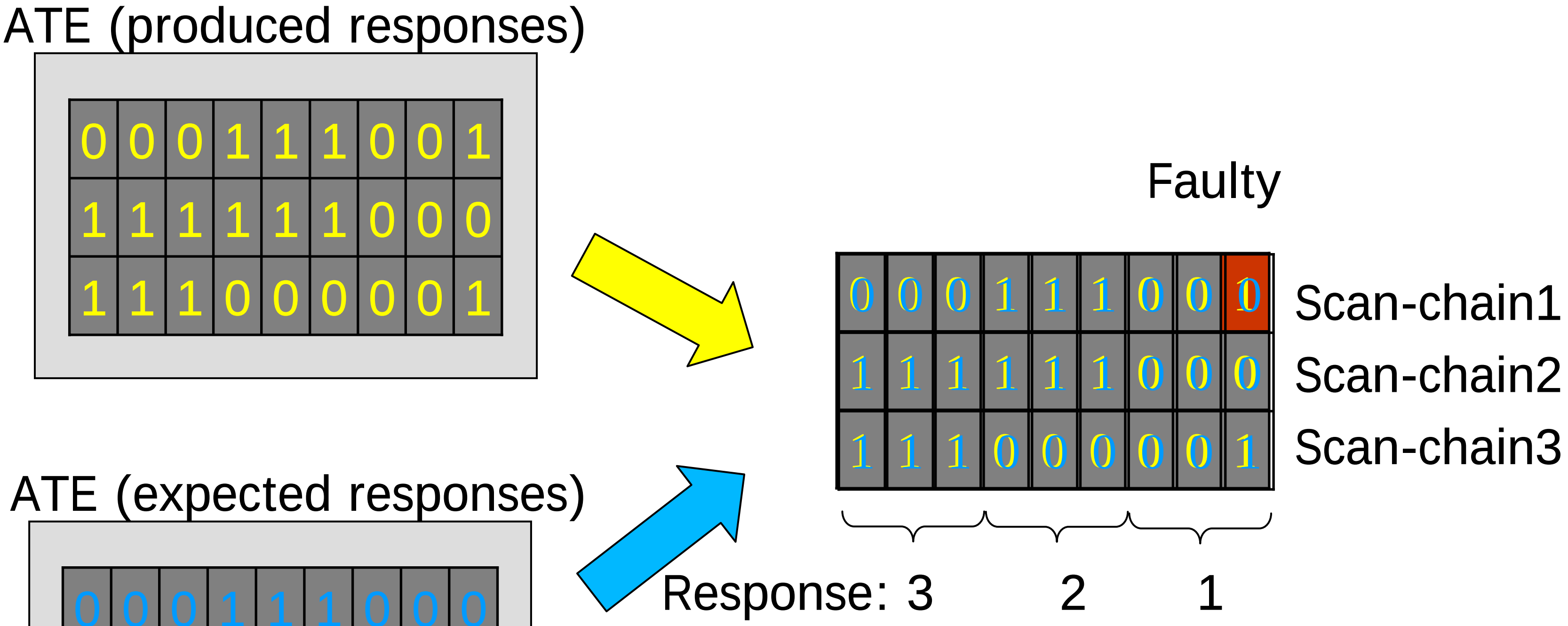
DUT

ATE (produced responses)

0	0	0	1	1	1	0	0	1
1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	0	0	1

1. Introduction

Testing



ATE (produced responses)

0	0	0	1	1	1	0	0	1
1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	0	0	1

ATE (expected responses)

0	0	0	1	1	1	0	0	0
1	1	1	1	1	1	0	0	0
1	1	1	0	0	0	0	0	1

Faulty

0	0	0	1	1	1	0	0	1 0	Scan-chain1
1	1	1	1	1	1	0	0	0	Scan-chain2
1	1	1	0	0	0	0	0	1	Scan-chain3

Response: 3 2 1

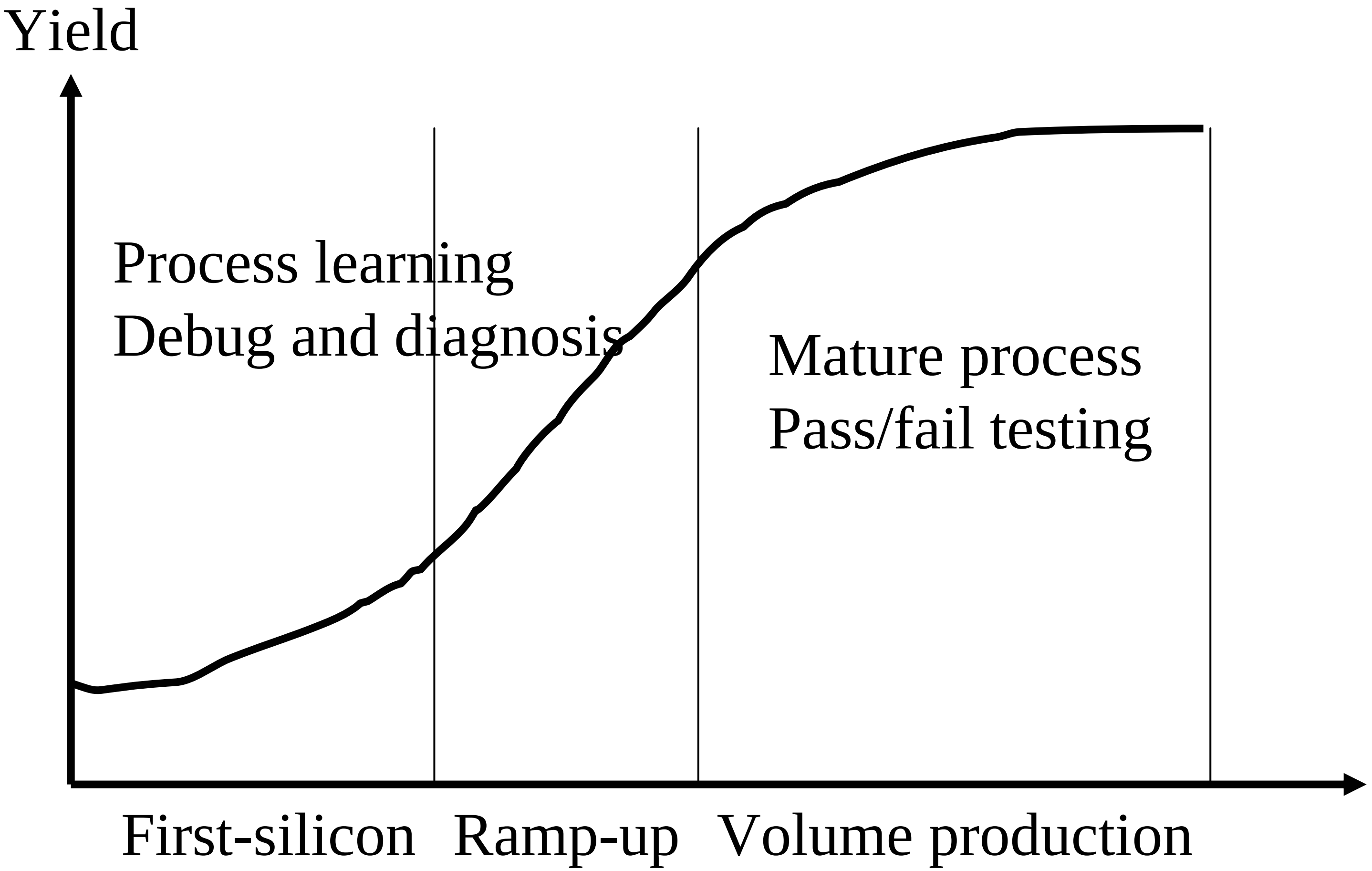
Fault detected by vector 1.
Fault at flip-flop 3 in scan-chain 1

Good for diagnosis

Outline

1. Introduction
2. Prior Work
    1. Multi-site test
    2. Abort-on-fail test
    3. Test data compression
3. Test Architecture
4. Experimental Results
5. Conclusion

Multi-site test



Yield

Process learning

Debug and diagnosis

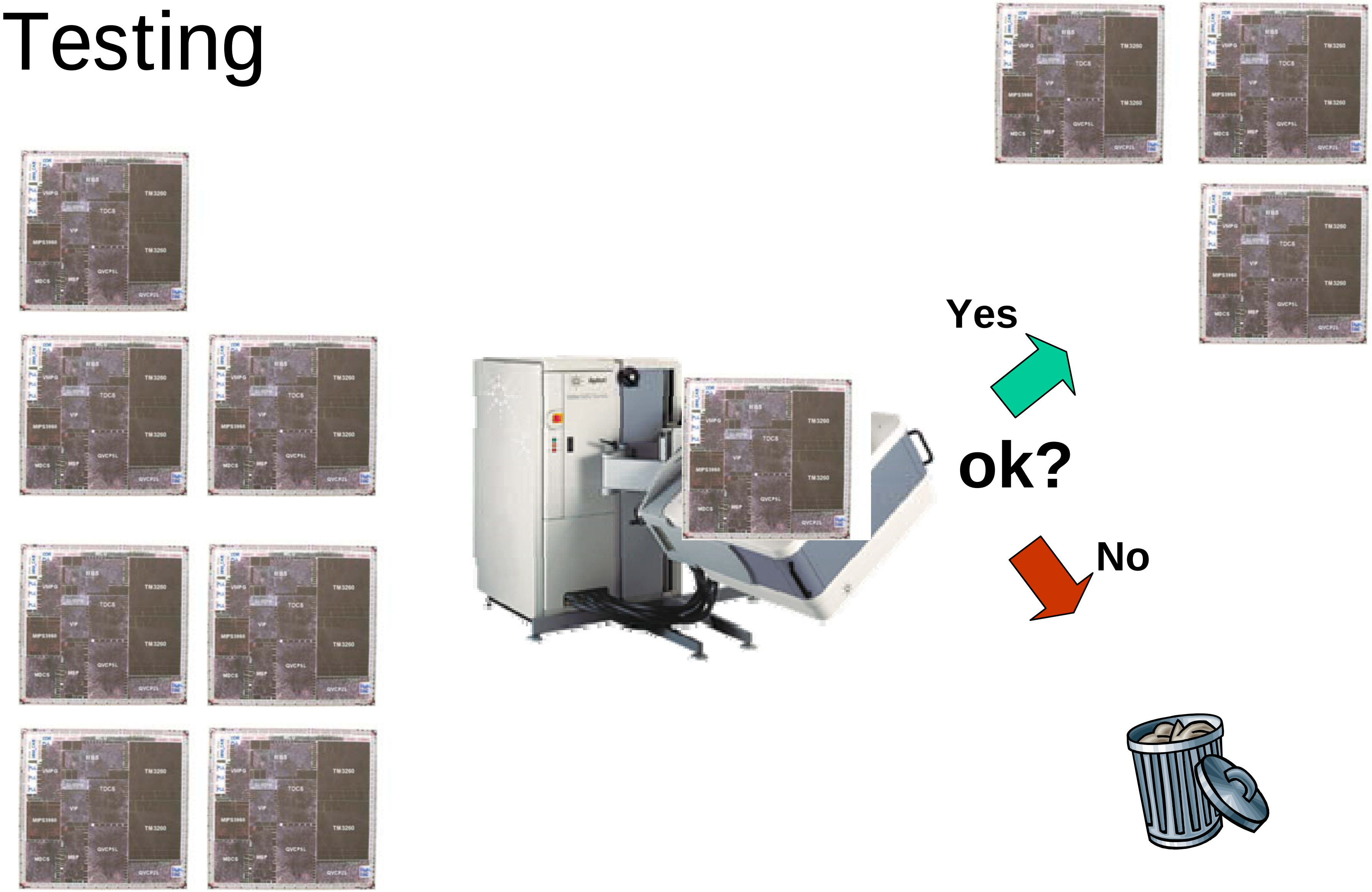
Mature process

Pass/fail testing

First-silicon Ramp-up Volume production

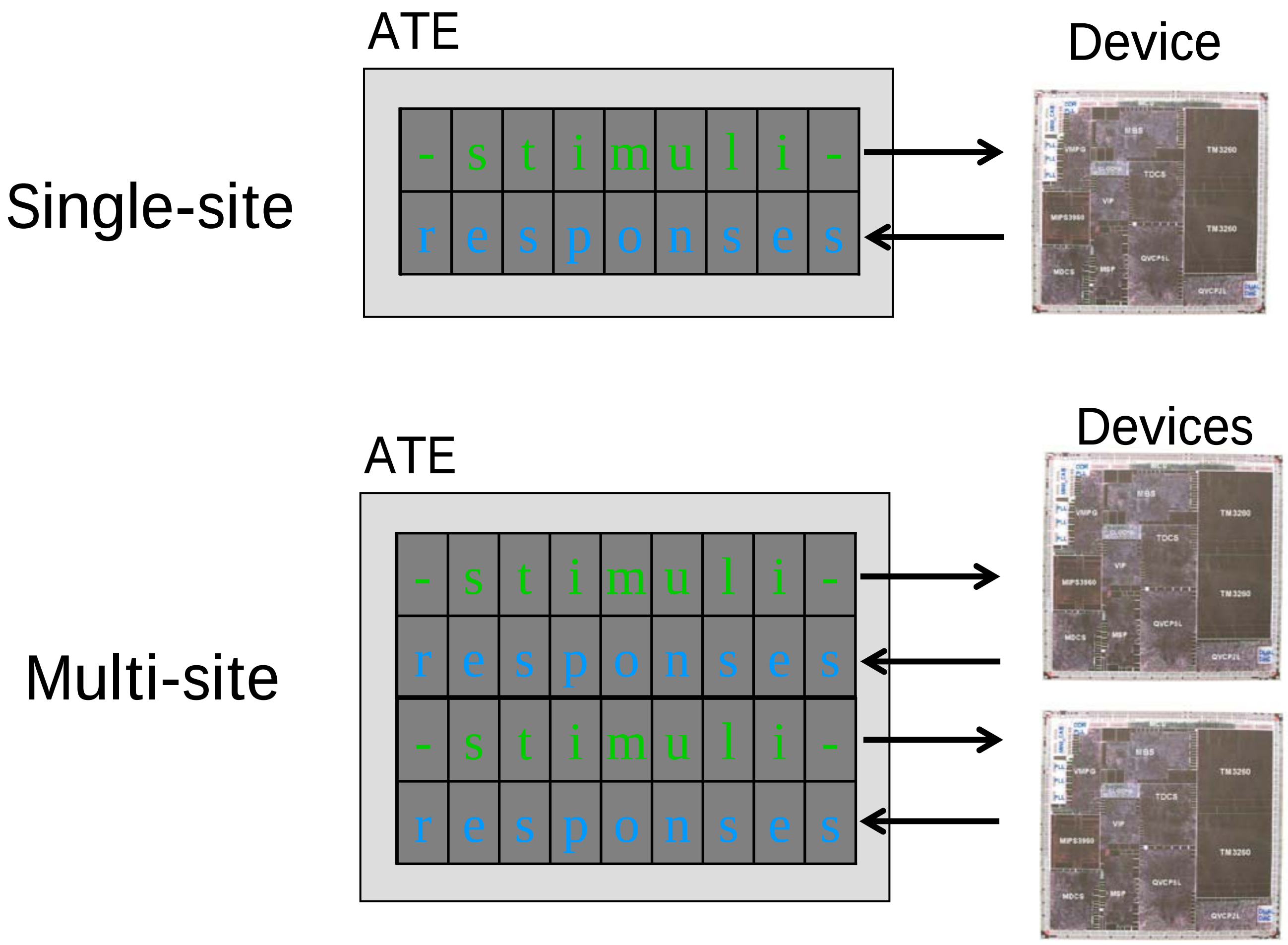
2. Prior Work

Testing



2. Prior Work

Multi-site Testing



Single-site

ATE

-stimuli-

responses

Device

Multi-site

ATE

-stimuli-

responses

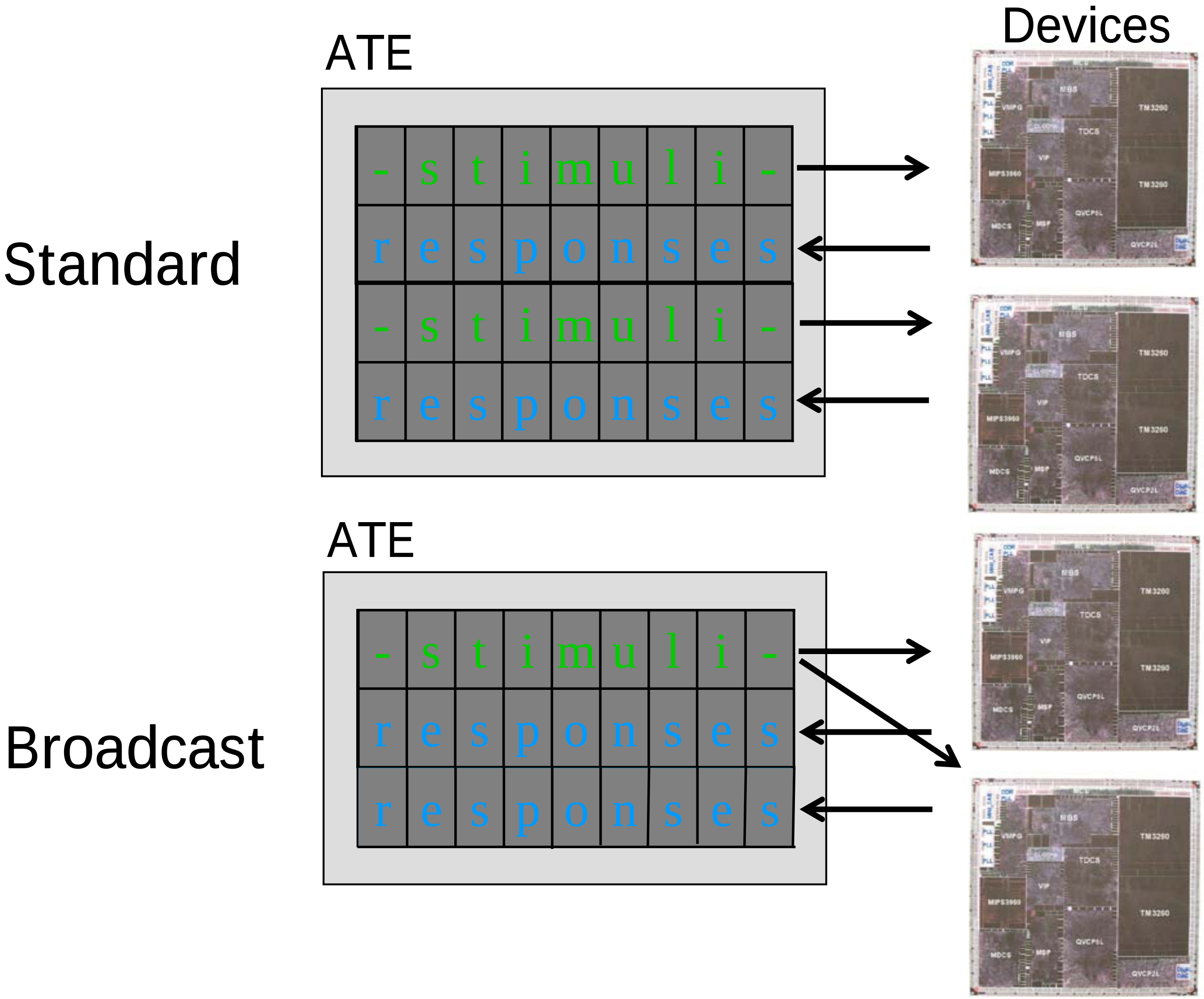
-stimuli-

responses

Devices

2. Prior Work

Multi-site Testing



Standard

ATE

- s t i m u l i -
r e s p o n s e s
- s t i m u l i -
r e s p o n s e s

Devices

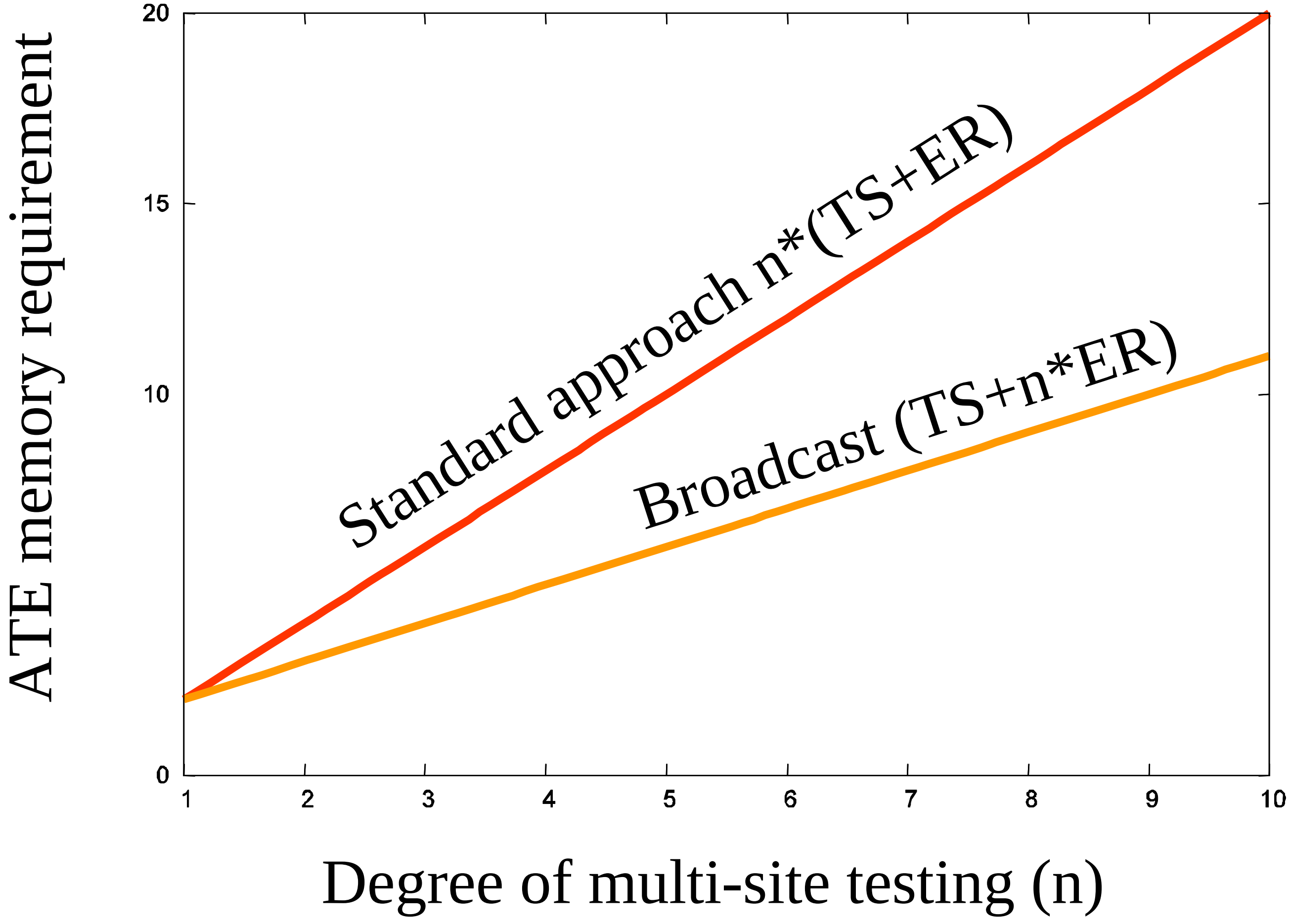
Broadcast

ATE

- s t i m u l i -
r e s p o n s e s
r e s p o n s e s

2. Prior Work

Multi-site Testing



Purpose

Problems when testing ICs:

- Long test times
- High ATE memory requirement
- Low throughput

+ **Multi-site testing** increases throughput

- Requires ATE memory

+ **Abort-on-fail testing** lower testing times

- Test data volume remains large

+ **Test data compression** lower ATE memory

- MISRs cannot terminate immediately when a fault is detected

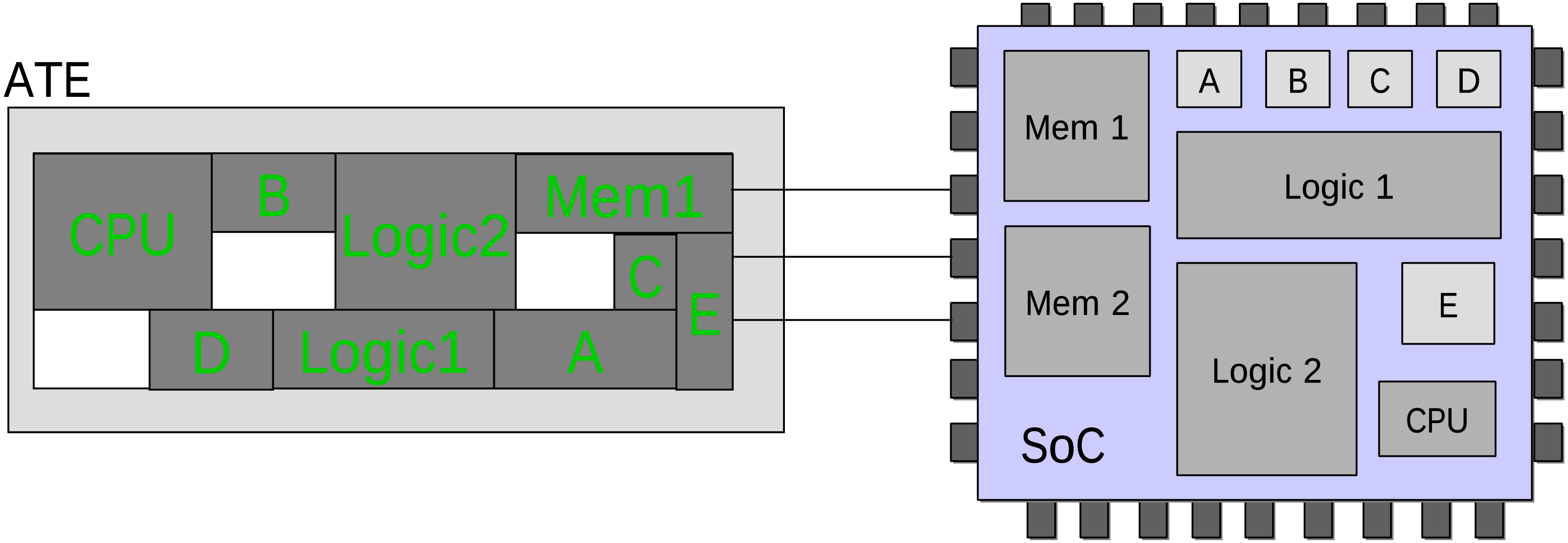
- Unknowns (X) must be handled

Aim: Define an architecture that allows:

1. High degree of multi-site testing
2. Test data volume compression, and
3. Abort-on-Fail testing

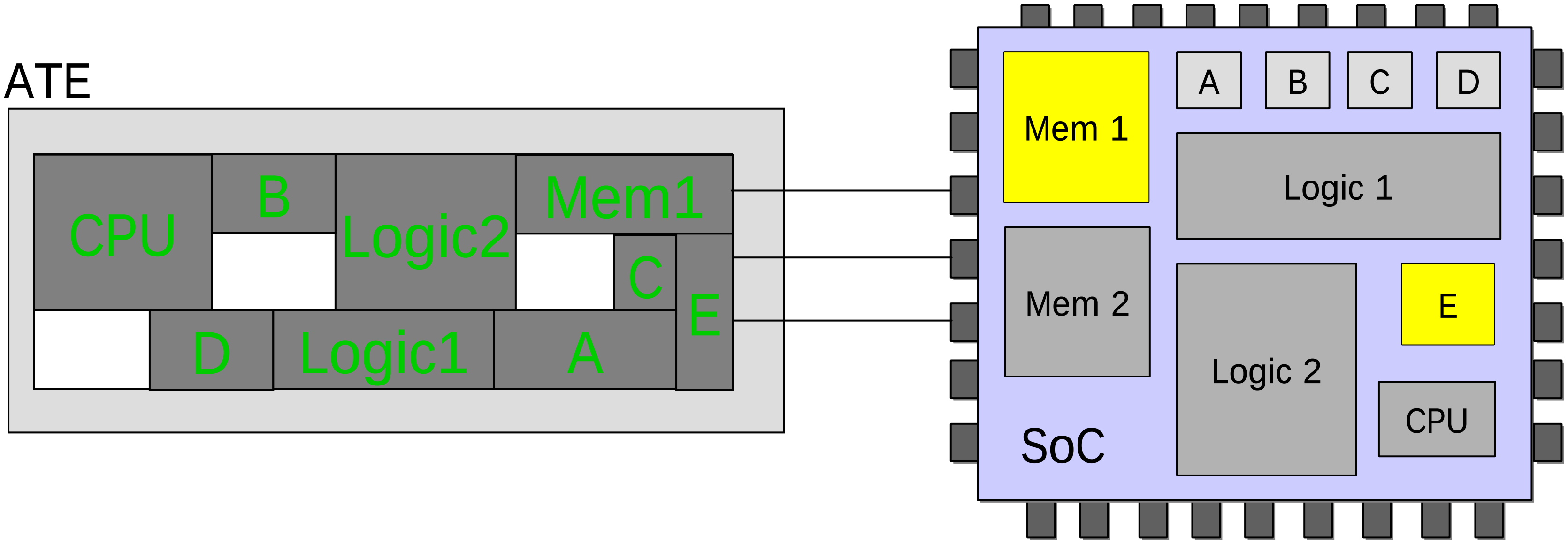
2. Prior Work

Test Scheduling



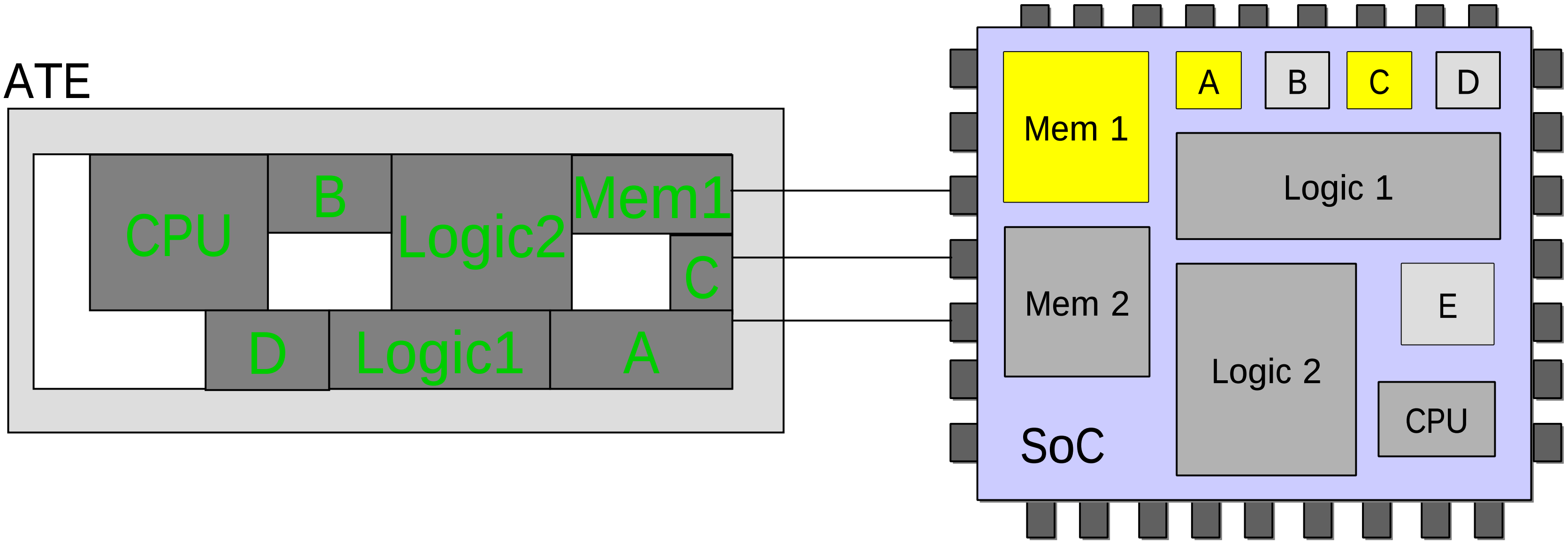
2. Prior Work

Test Scheduling



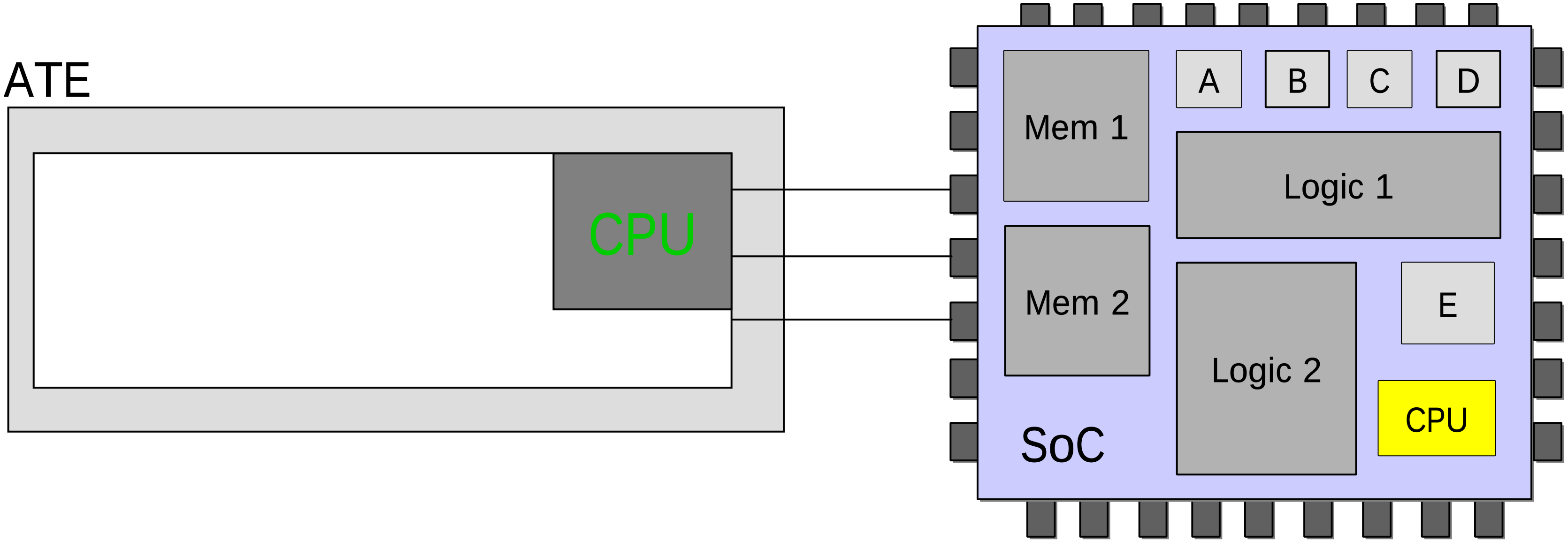
2. Prior Work

Test Scheduling



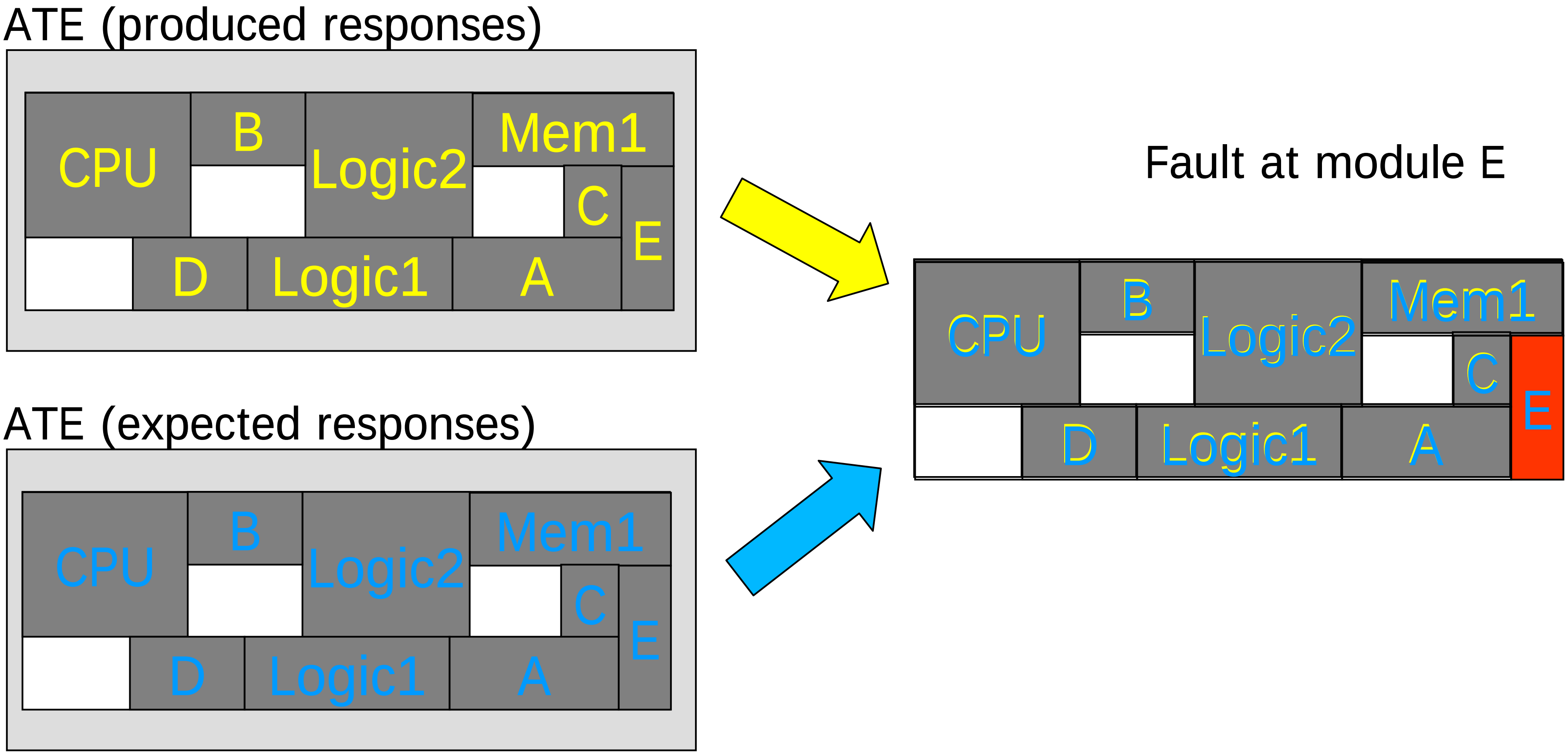
2. Prior Work

Test Scheduling



2. Prior Work

Test Scheduling



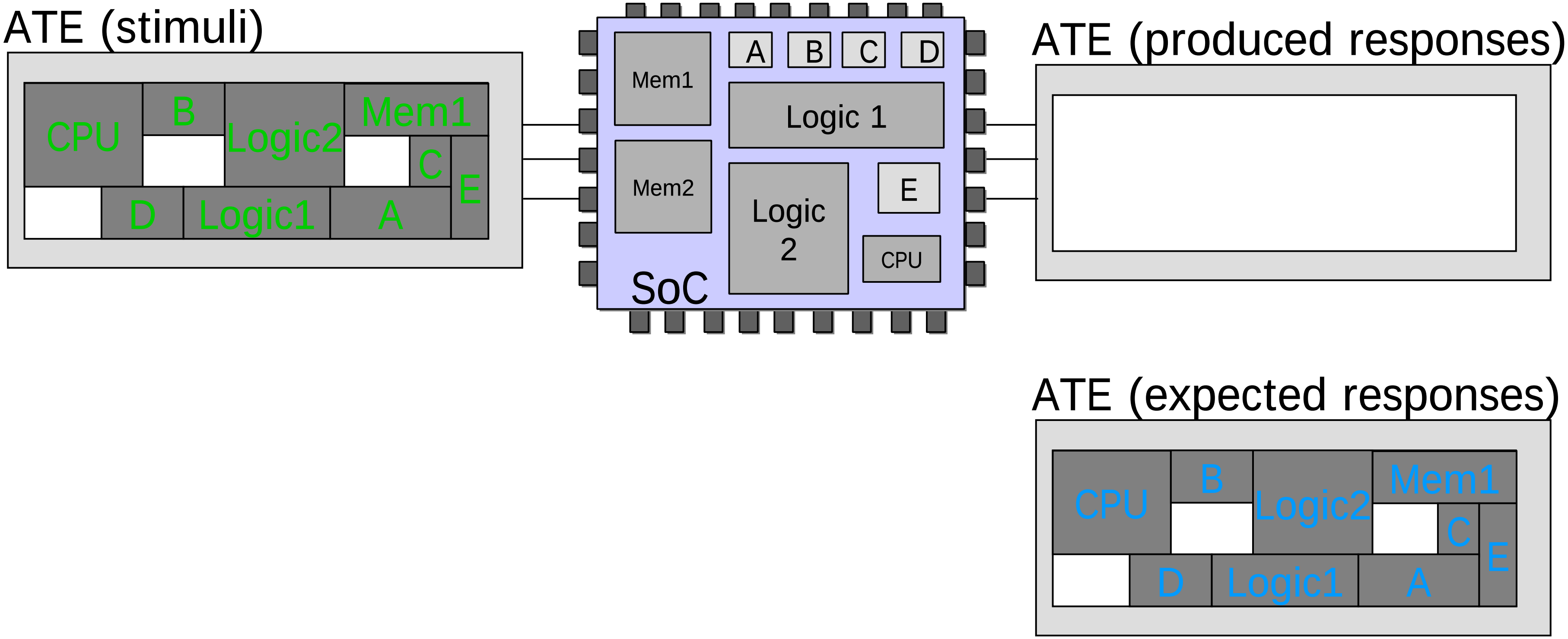
ATE (produced responses)

ATE (expected responses)

Fault at module E

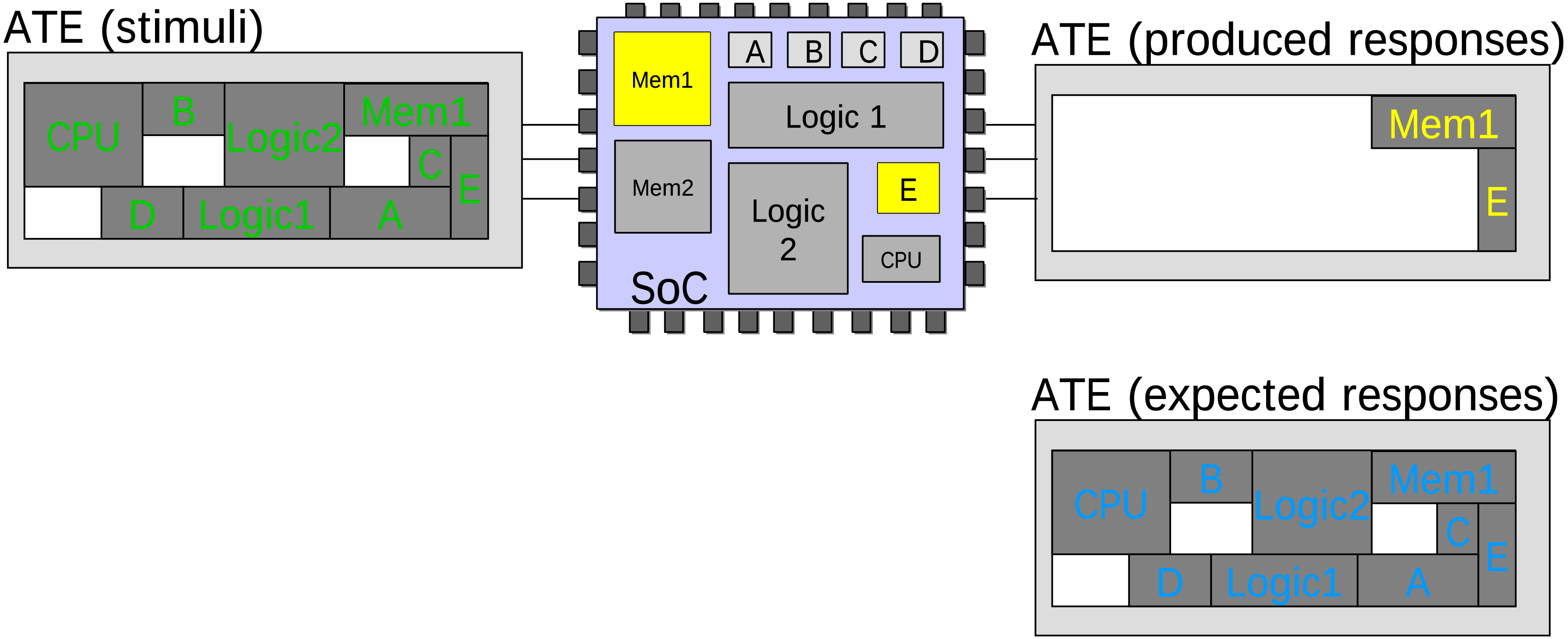
2. Prior Work

Abort-on-Fail Testing



2. Prior Work

Abort-on-Fail Testing

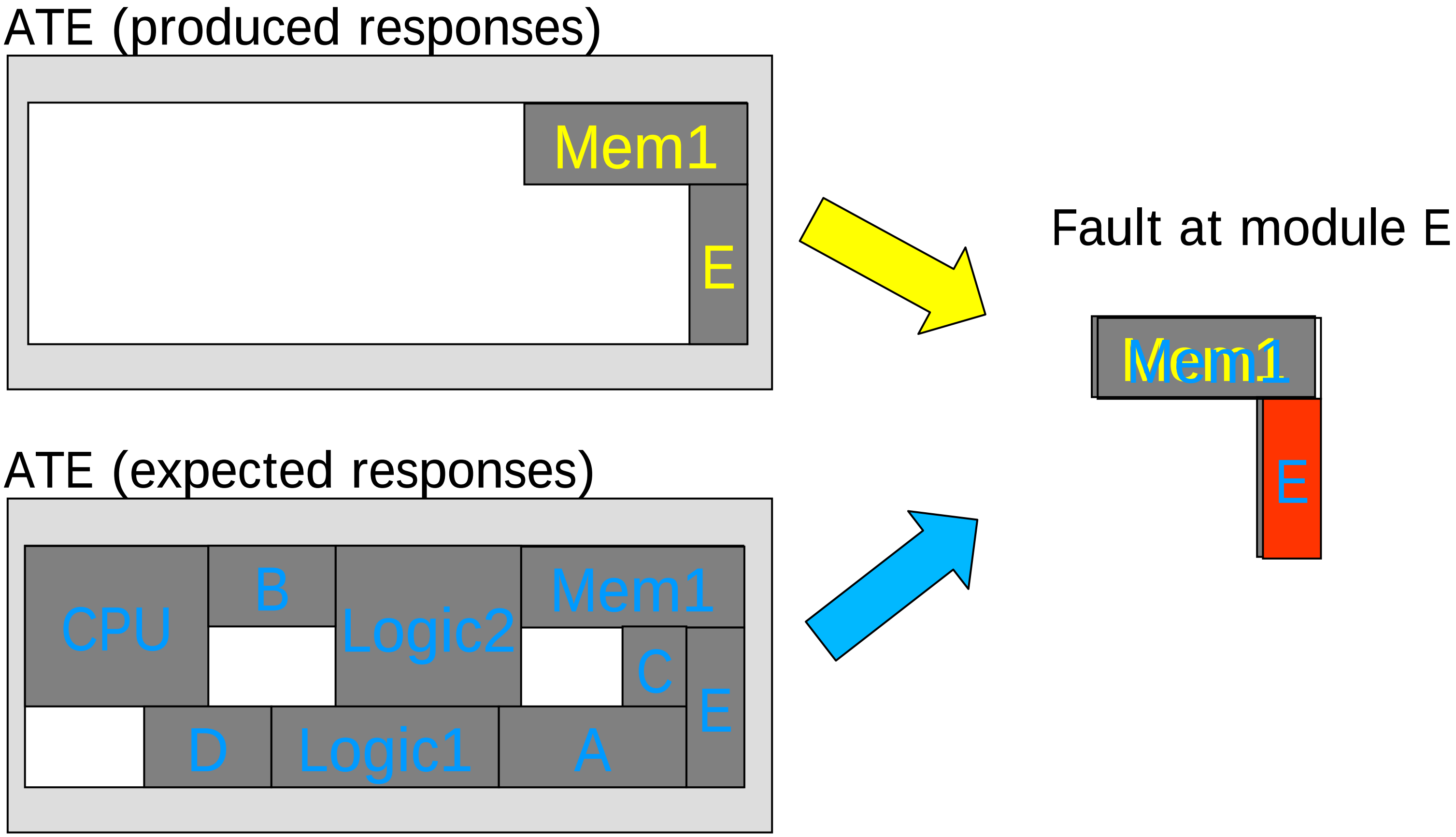

ATE (stimuli)

ATE (produced responses)

ATE (expected responses)

2. Prior Work

Abort-on-Fail Testing



ATE (produced responses)

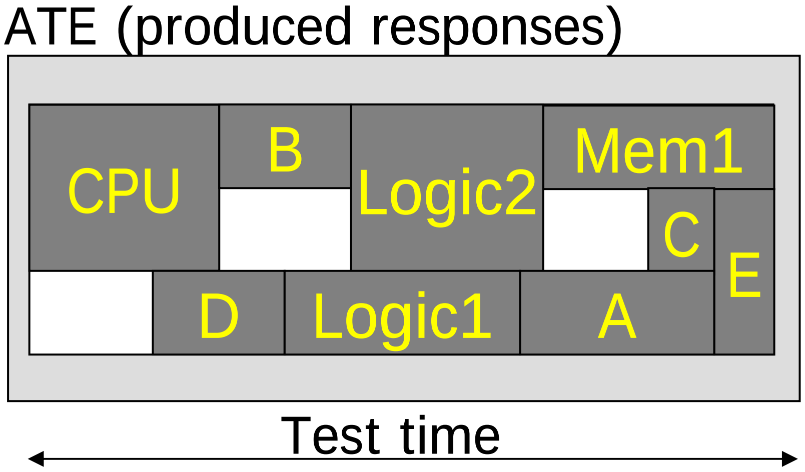
ATE (expected responses)

Fault at module E

2. Prior Work

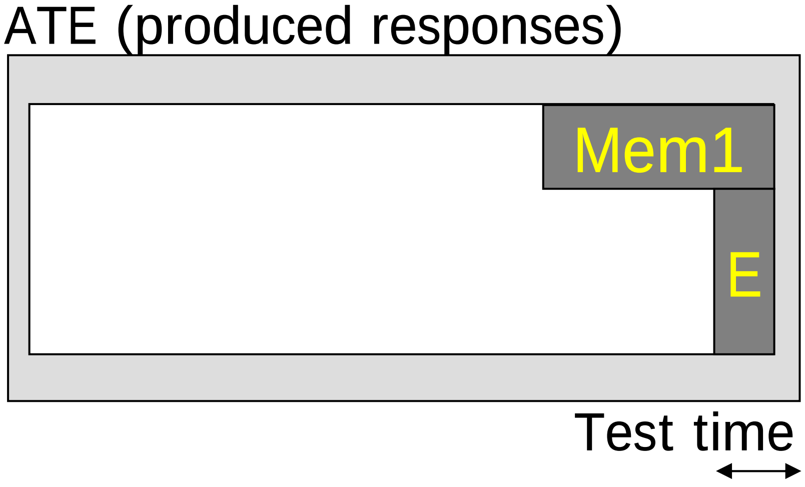
Abort-on-Fail Testing

Without Abort-on-Fail



Time to determine a possible fault in module E

With Abort-on-Fail



2. Prior Work

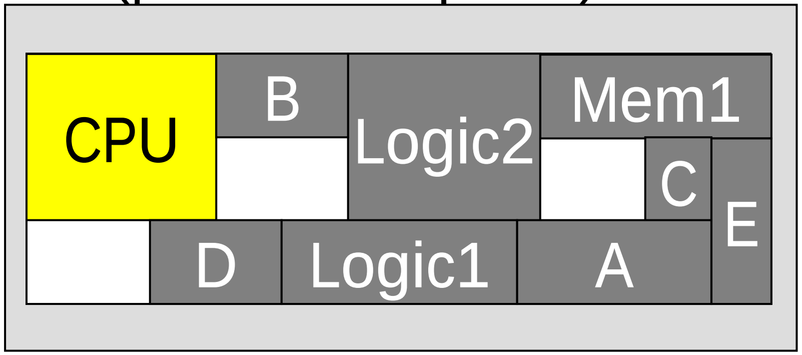
Abort-on-Fail Testing

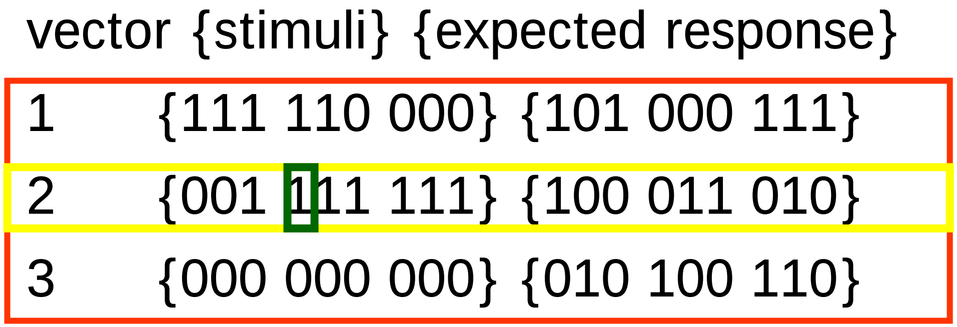
- Spend less time on faulty circuits
- If the test fails, testing is aborted early
- Low-yielding and short tests should be performed early

2. Prior Work

Abortion Granularity

ATE (produced response)





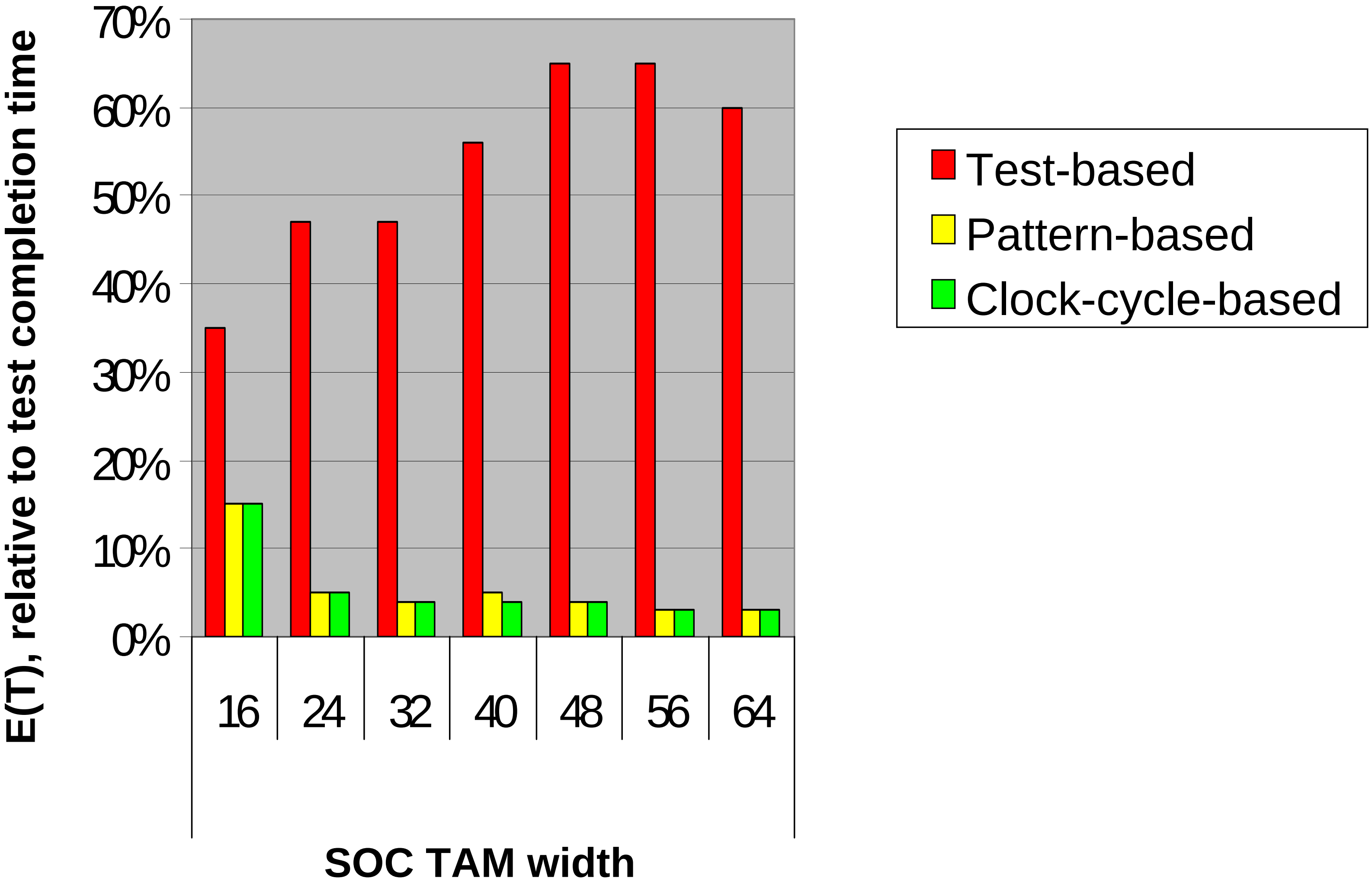
vector	{stimuli}	{expected response}
1	{111 110 000}	{101 000 111}
2	{001 111 111}	{100 011 010}
3	{000 000 000}	{010 100 110}

Possible termination

- Per test
- Per vector
- Per clock cycle

2. Prior Work

Abortion Granularity



Purpose

Problems when testing ICs:

- Long test times
- High ATE memory requirement
- Low throughput

+ **Multi-site testing** increases throughput

- Requires ATE memory

+ **Abort-on-fail testing** lower testing times

- Test data volume remains large

+ **Test data compression** lower ATE memory

- MISRs cannot terminate immediately when a fault is detected

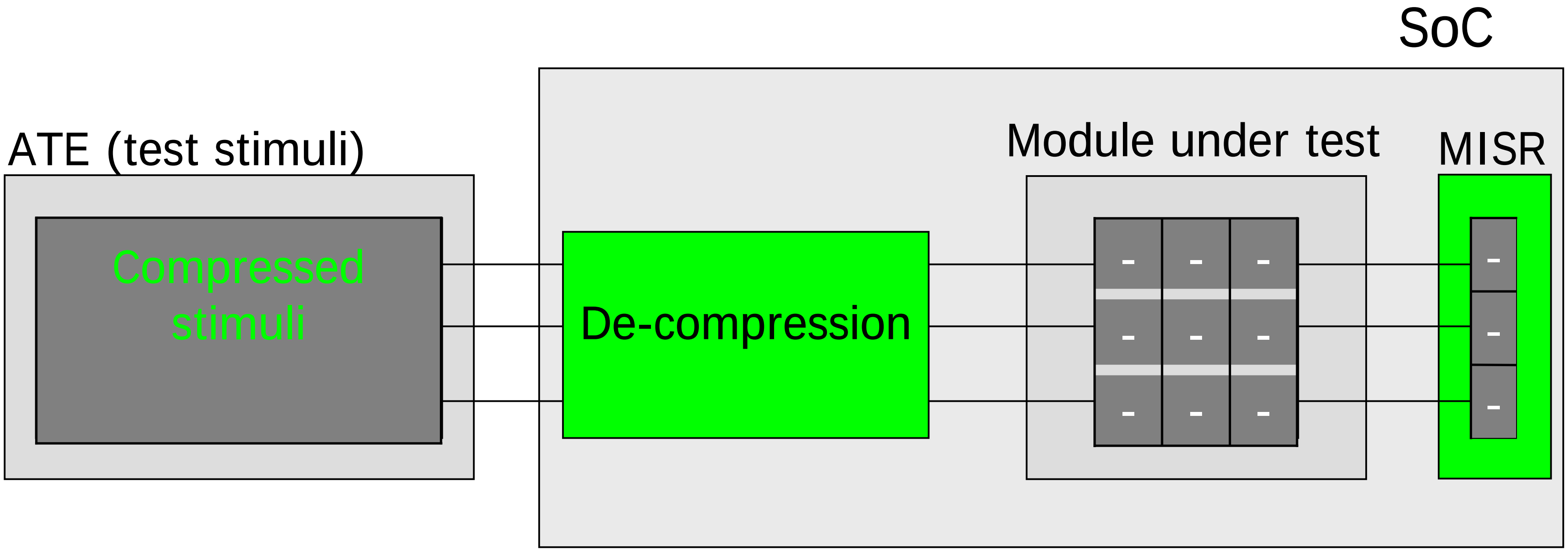
- Unknowns (X) must be handled

Aim: Define an architecture that allows:

1. High degree of multi-site testing
2. Test data volume compression, and
3. Abort-on-Fail testing

2. Prior Work

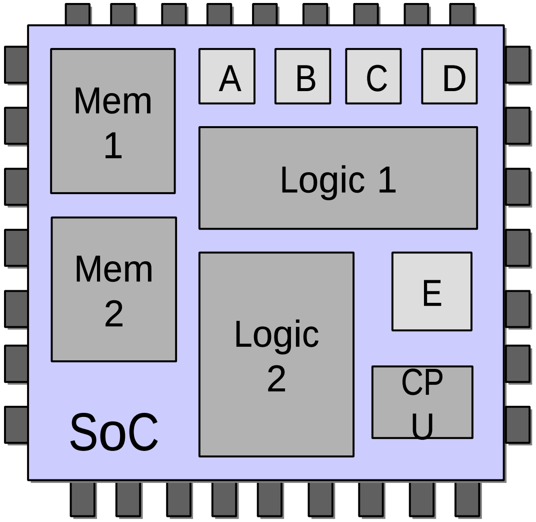
Test Data Compression



Test Data Compression

For all modules

- **Fill test stimuli don’t care bits**
- Simulate to find test responses
- Compress test stimuli
- Design de-compression logic
- Design MISR



vector	{stimuli}	{expected response}
1	{1xx xx0 xxx}	{x0x xxx x11}
2	{xx1 xx1 xxx}	{1xx 0xx xxx}
3	{0x0 xxx xxx}	{xxx 1xx xxx}

vector	{stimuli}	{expected response}
1	{111 110 000}	{x0x xxx x11}
2	{001 111 111}	{1xx 0xx xxx}
3	{000 000 000}	{xxx 1xx xxx}

Test Data Compression

For all modules

- Fill test stimuli don’t care bits
- **Simulate to find test responses**
- Compress test stimuli
- Design de-compression logic
- Design MISR

vector	{stimuli}	{expected response}
1	{111 110 000}	{x0x xxx x11}
2	{001 111 111}	{1xx 0xx xxx}
3	{000 000 000}	{xxx 1xx xxx}

vector	{stimuli}	{expected response}
1	{111 110 000}	{101 100 011}
2	{001 111 111}	{100 010 010}
3	{000 000 000}	{010 101 100}

Test Data Compression

For all modules

- Fill test stimuli don’t care bits
- Simulate to find test responses
- **Compress test stimuli**
- Design de-compression logic
- Design MISR

vector	{stimuli}	{expected response}
1	{111 110 000}	{x0x xxx x11}
2	{001 111 111}	{1xx 0xx xxx}
3	{000 000 000}	{xxx 1xx xxx}

→

{compressed stimuli}

{001 010 101}

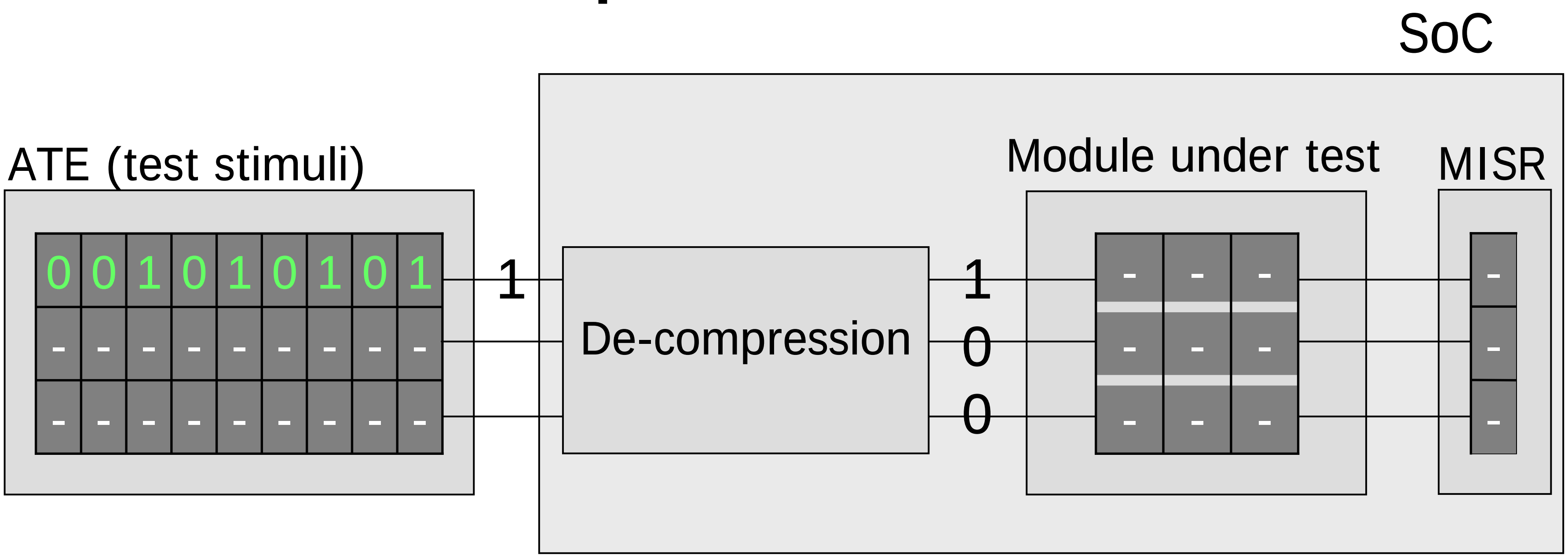
Test Data Compression

For all modules

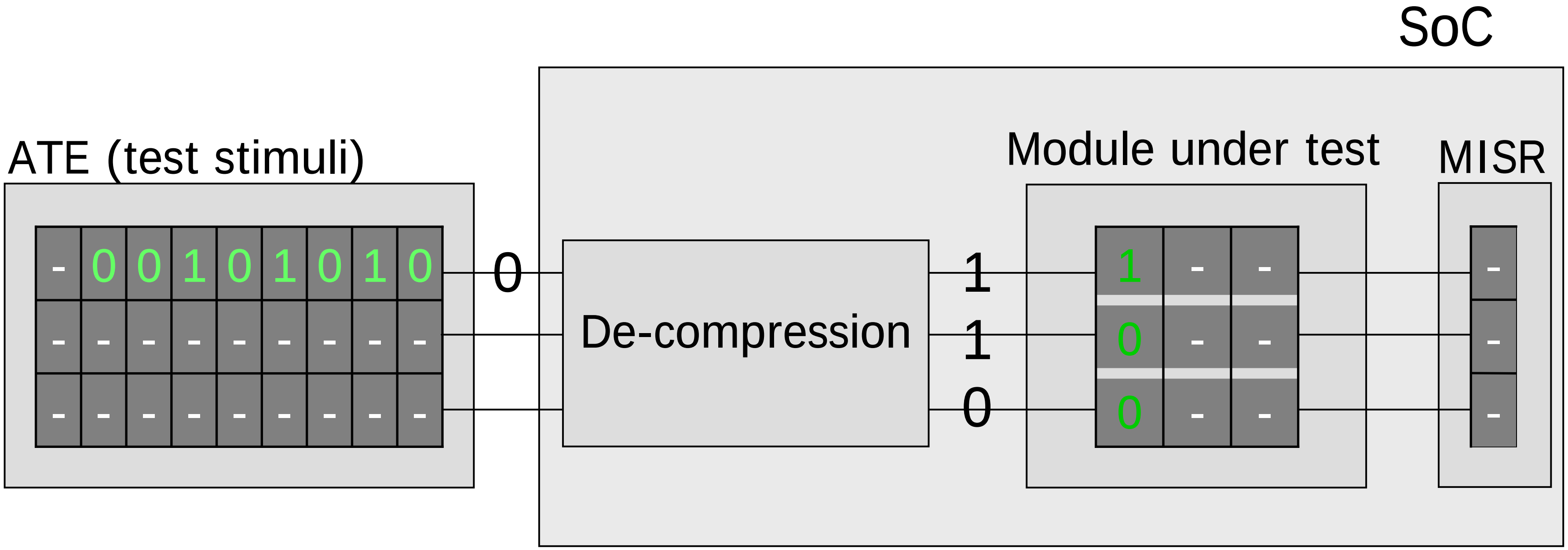
- Fill test stimuli don’t care bits
- Simulate to find test responses
- Compress test stimuli
- **Design de-compression logic**
- **Design MISR**

vector	{stimuli}	{expected response}		{compressed stimuli}
1	{111 110 000}	{x0x xxx x11}	→	{001 010 101}
2	{001 111 111}	{1xx 0xx xxx}		
3	{000 000 000}	{xxx 1xx xxx}		

Test Data Compression



Test Data Compression



SoC

ATE (test stimuli)

-	0	0	1	0	1	0	1	0
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

0

De-compression

1
1
0

Module under test

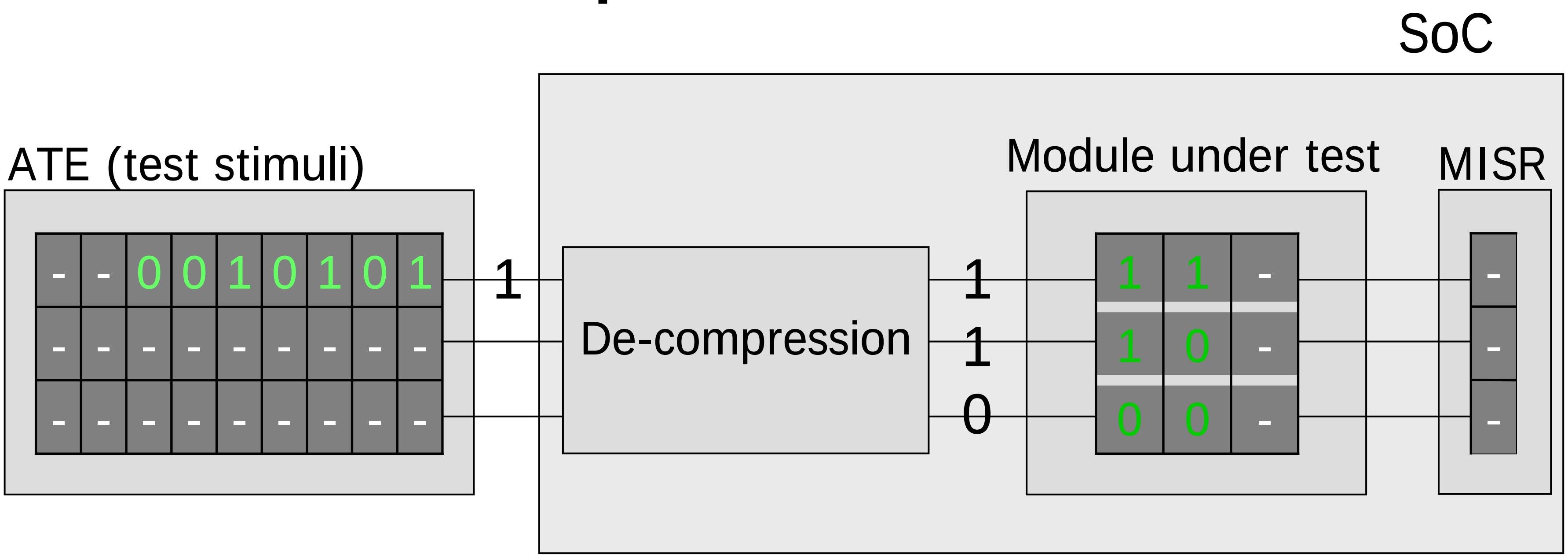
1	-	-
0	-	-
0	-	-

MISR

-
-
-

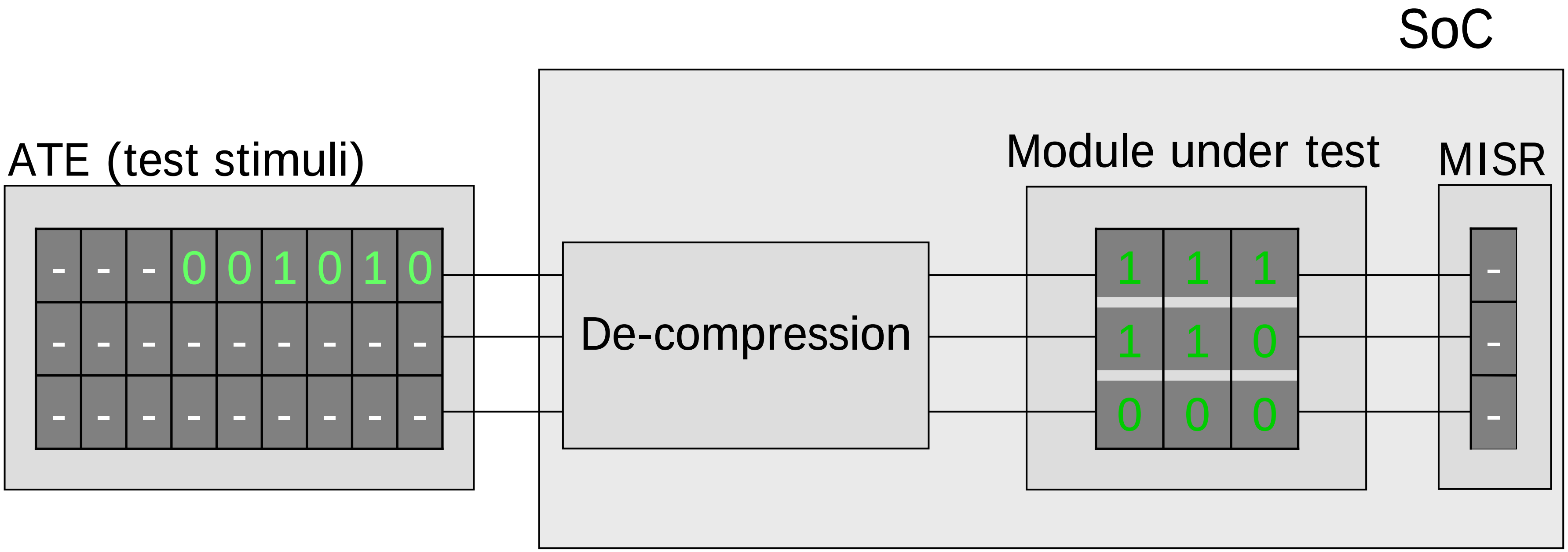
2. Prior Work

Test Data Compression



2. Prior Work

Test Data Compression



SoC

ATE (test stimuli)

-	-	-	0	0	1	0	1	0
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

De-compression

Module under test

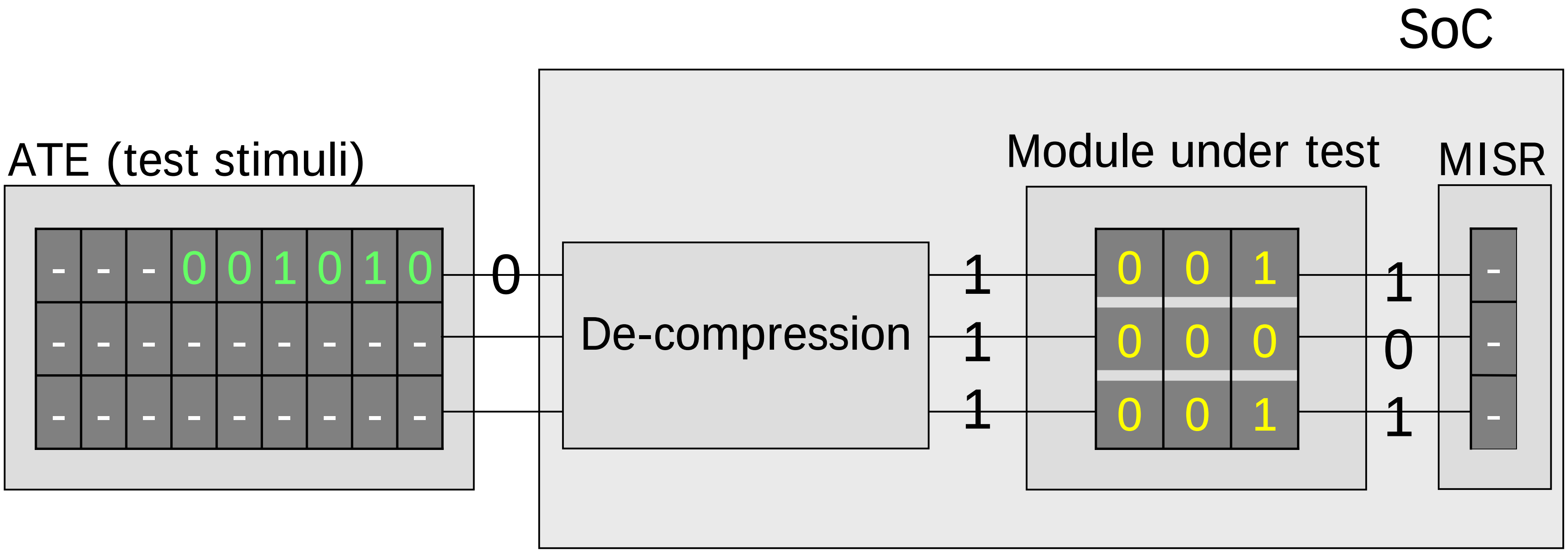
1	1	1
1	1	0
0	0	0

MISR

-
-
-

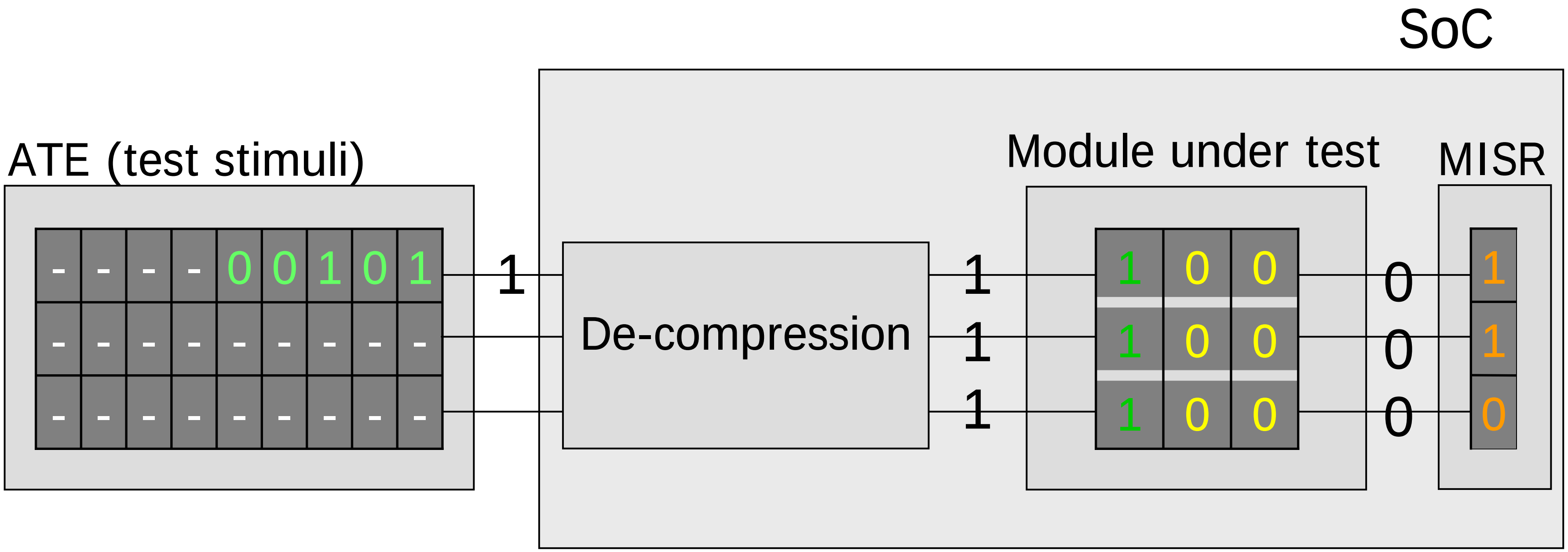
2. Prior Work

Test Data Compression

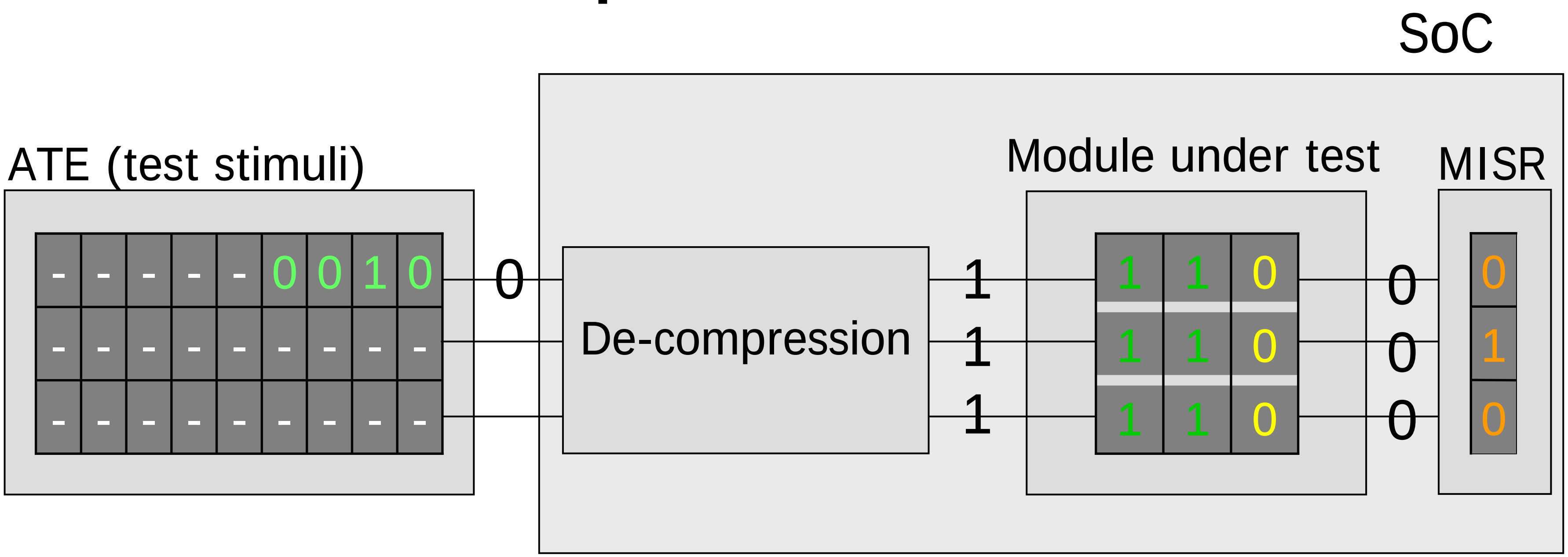


2. Prior Work

Test Data Compression

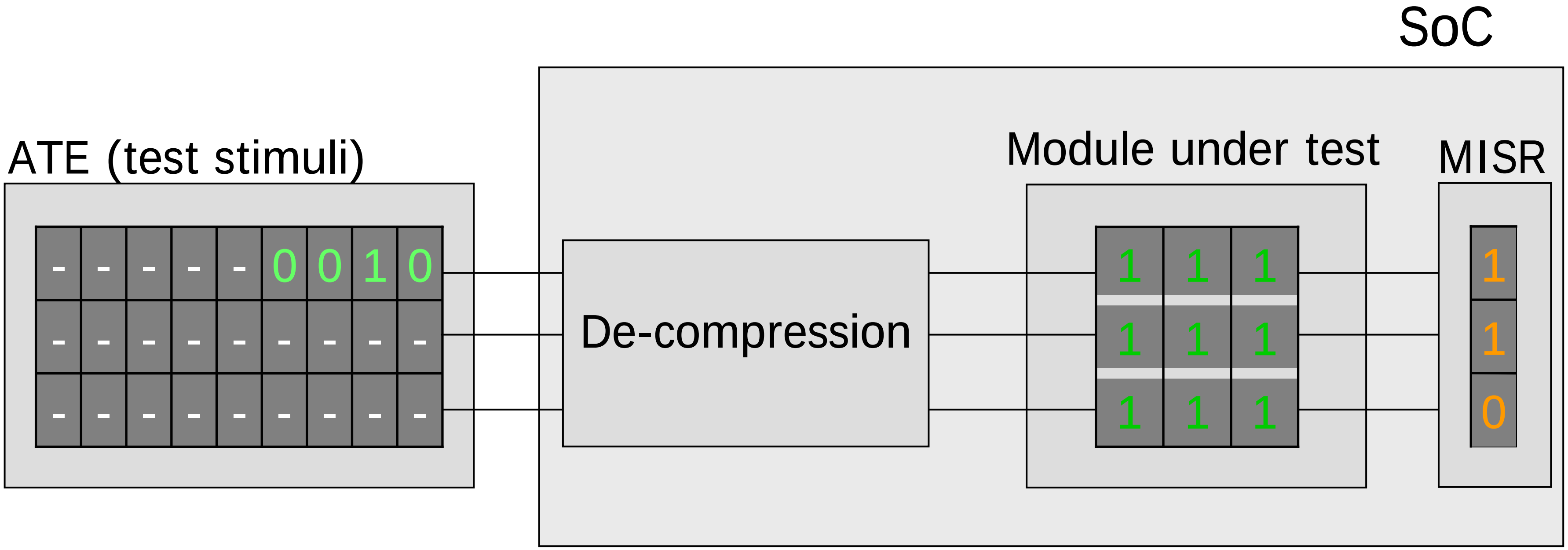


Test Data Compression

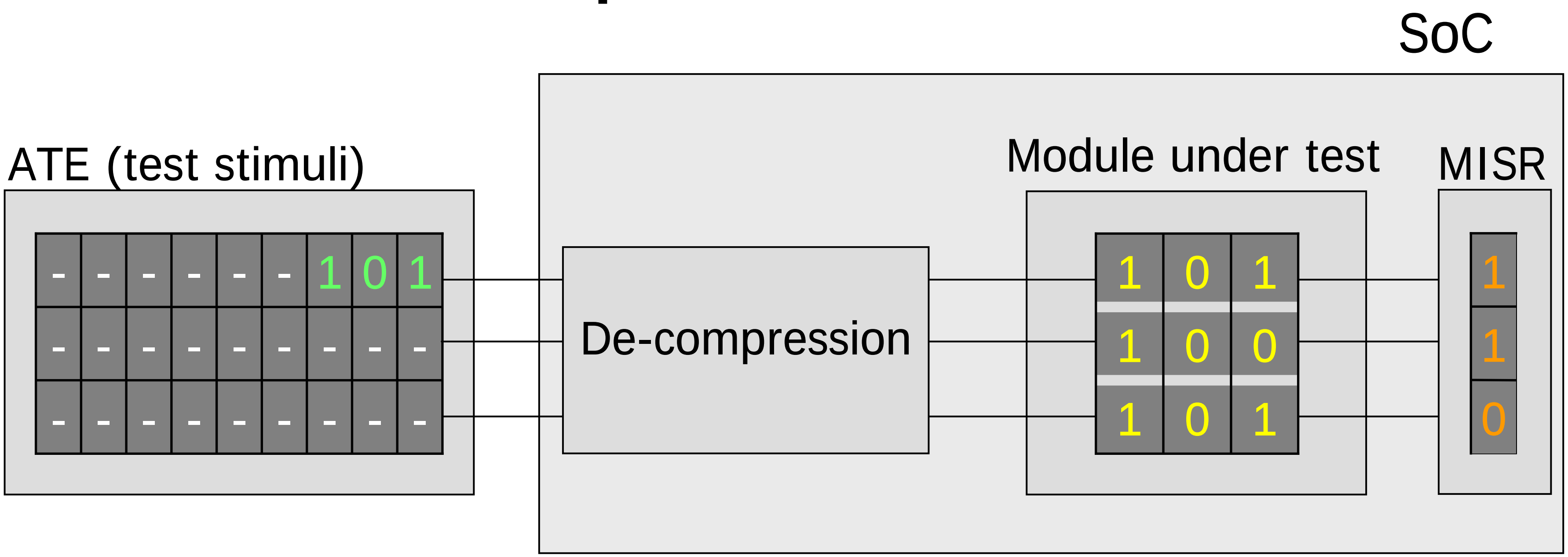


2. Prior Work

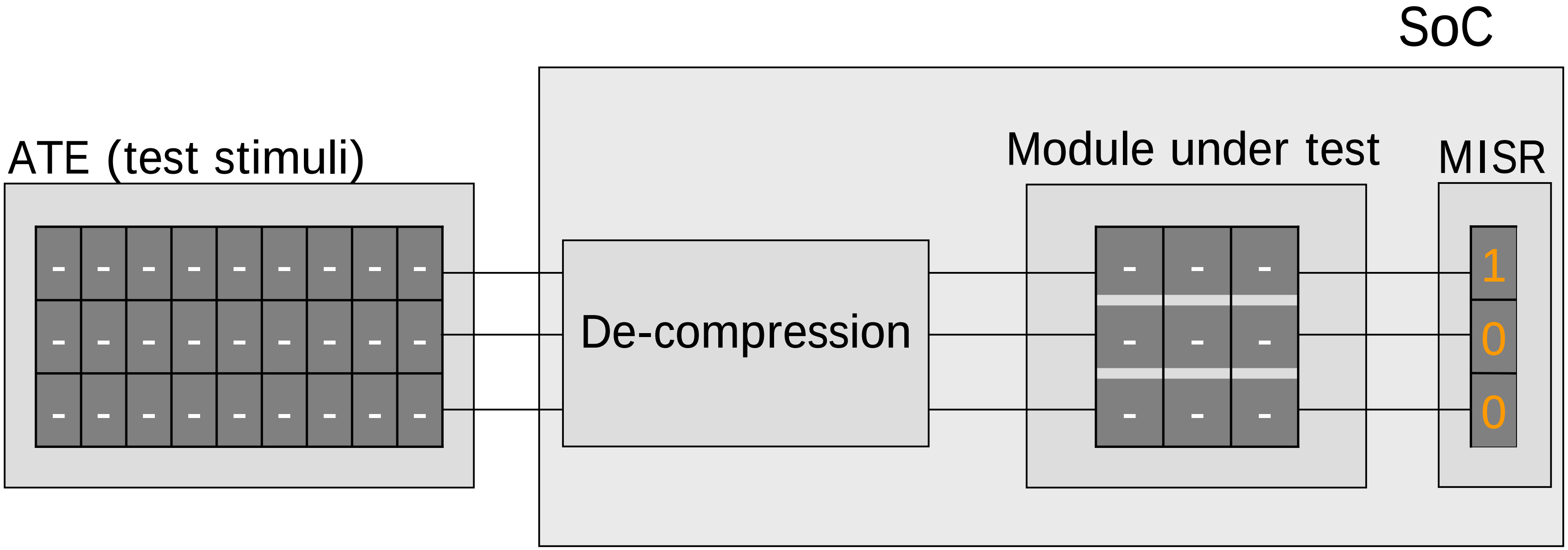
Test Data Compression



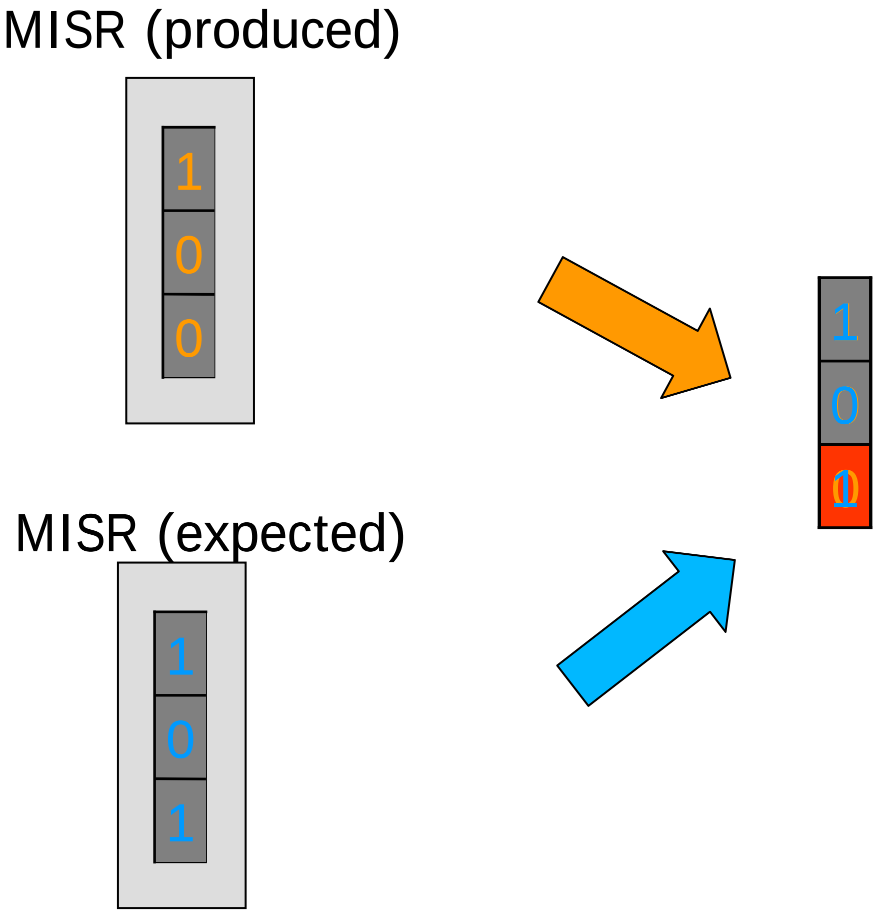
Test Data Compression



Test Data Compression



Test Data Compression



MISR (produced)

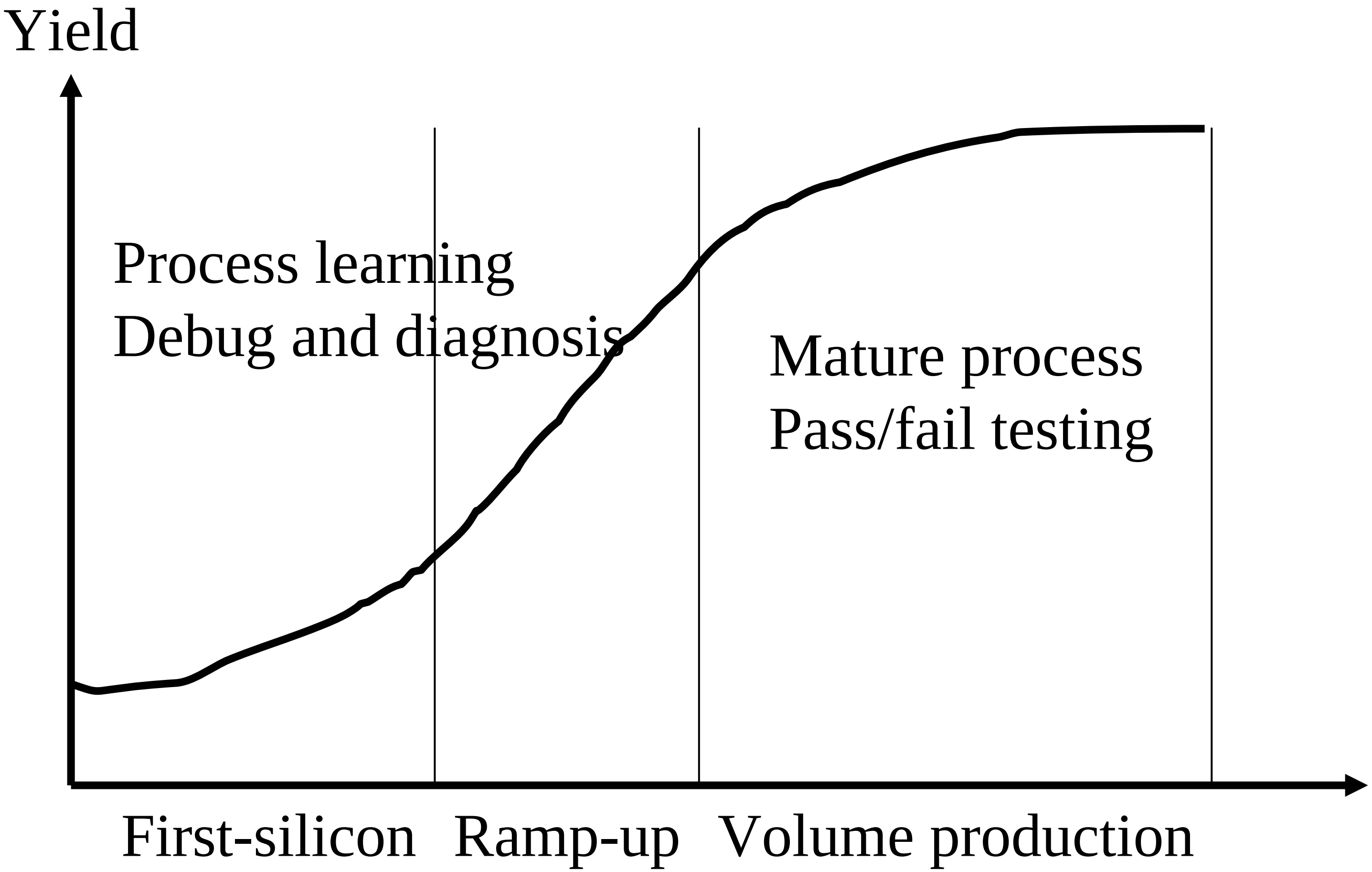
MISR (expected)

Faulty module!

But where is the fault?

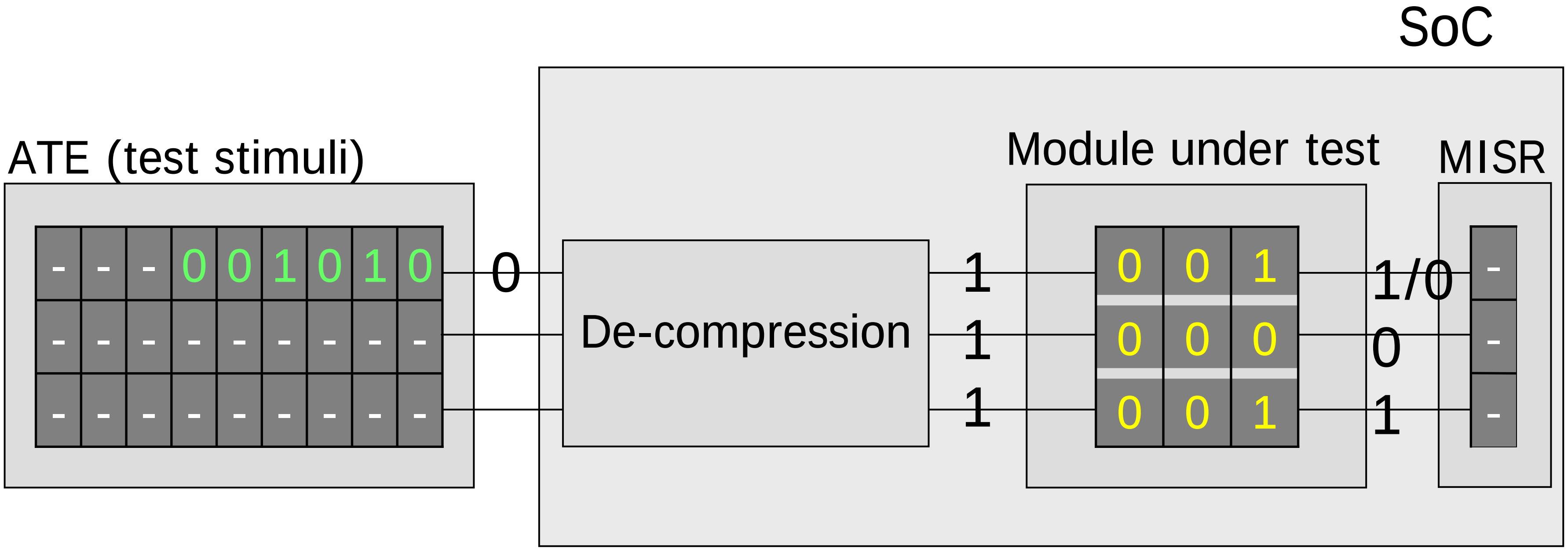
2. Prior Work

Test Data Compression



2. Prior Work

Unknowns (X)



x
Expected response: 0
1

Purpose

Problems when testing ICs:

- Long test times
- High ATE memory requirement
- Low throughput

+ **Multi-site testing** increases throughput

- Requires ATE memory

+ **Abort-on-fail testing** lower testing times

- Test data volume remains large

+ **Test data compression** lower ATE memory

- MISRs cannot terminate immediately when a fault is detected

- Unknowns (X) must be handled

Aim: Define an architecture that allows:

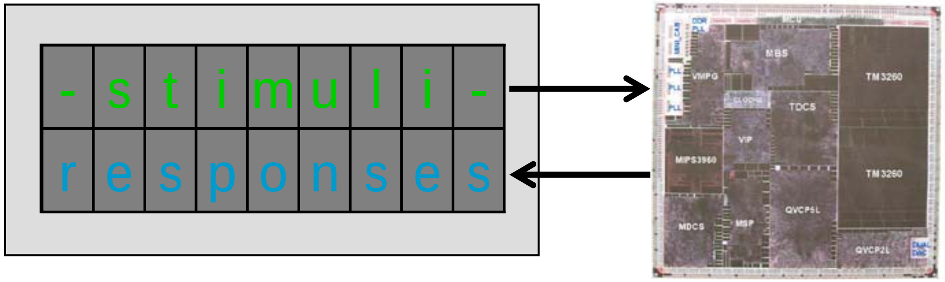
1. High degree of multi-site testing
2. Test data volume compression, and
3. Abort-on-Fail testing

Outline

1. Introduction
2. Prior Work
3. Test Architecture
4. Experimental Results
5. Conclusion

3. Test Architecture

Approach



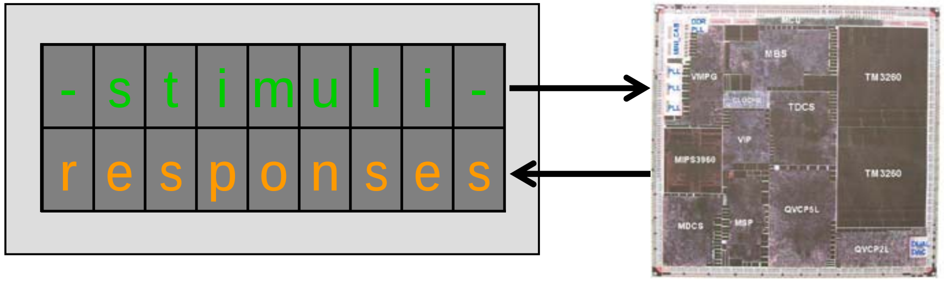
- s t i m u l i -

r e s p o n s e s

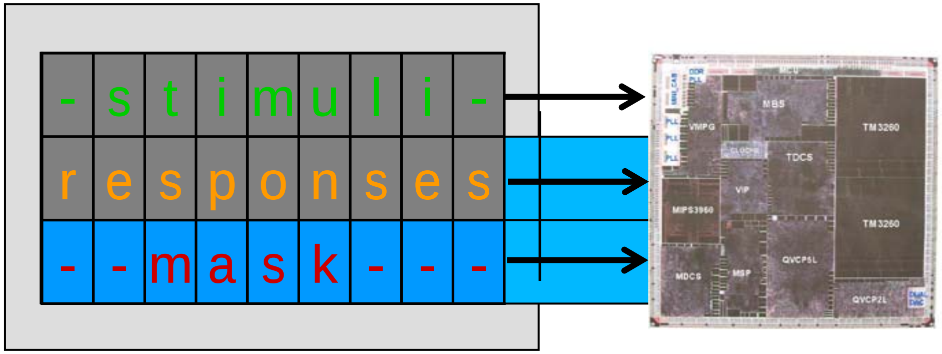
3. Test Architecture

Approach

Prior approach:



Proposed approach:



3. Test Architecture

Approach

For all modules

- **Define Mask**
- Fill all don’t care bits
- Compress test stimuli, expected responses, and mask data
- Design de-compression logic

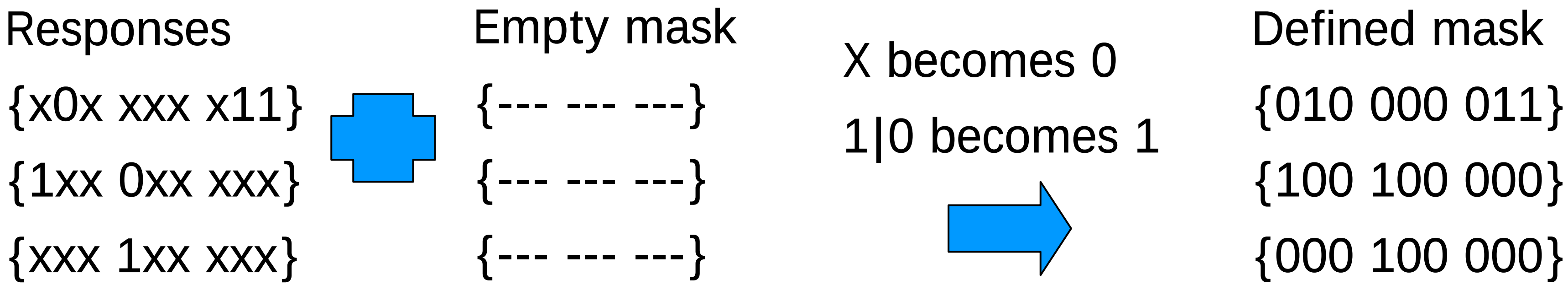
vector	{stimuli}	{expected responses}		empty mask
1	{1xx xx0 xxx}	{x0x xxx x11}	+	{--- --- ---}
2	{xx1 xx1 xxx}	{1xx 0xx xxx}		{--- --- ---}
3	{0x0 xxx xxx}	{xxx 1xx xxx}		{--- --- ---}

3. Test Architecture

Approach

For all modules

- **Define Mask**
- Fill all don’t care bits
- Compress test stimuli, expected responses, and mask data
- Design de-compression logic



Responses

{x0x xxx x11}

{1xx 0xx xxx}

{xxx 1xx xxx}

Empty mask

{--- --- ---}

{--- --- ---}

{--- --- ---}

X becomes 0

1|0 becomes 1

Defined mask

{010 000 011}

{100 100 000}

{000 100 000}

Approach

For all modules

- Define Mask
- **Fill <u>all</u> don’t care bits**
- Compress test stimuli, expected responses, and mask data
- Design de-compression logic

vector	{stimuli}	{expected responses}		{stimuli}	{expected responses}
1	{1xx xx0 xxx}	{x0x xxx x11}	→	{111 110 000}	{000 000 111}
2	{xx1 xx1 xxx}	{1xx 0xx xxx}		{001 111 111}	{100 000 111}
3	{0x0 xxx xxx}	{xxx 1xx xxx}		{000 000 000}	{100 100 111}

3. Test Architecture

Approach

For all modules

- Define Mask
- Fill all don’t care bits
- **Compress test stimuli, expected responses, and mask data**
- Design de-compression logic

vector	{stimuli}	{expected responses}	{mask}
1	{111 110 000}	{000 000 111}	{010 000 011}
2	{001 111 111}	{100 000 111}	{100 100 000}
3	{000 000 000}	{100 100 111}	{000 100 000}

{compressed stimuli}

{001 010 101}

{compressed responses

{100 010 110}

{compressed mask}

{010 111 000}

3. Test Architecture

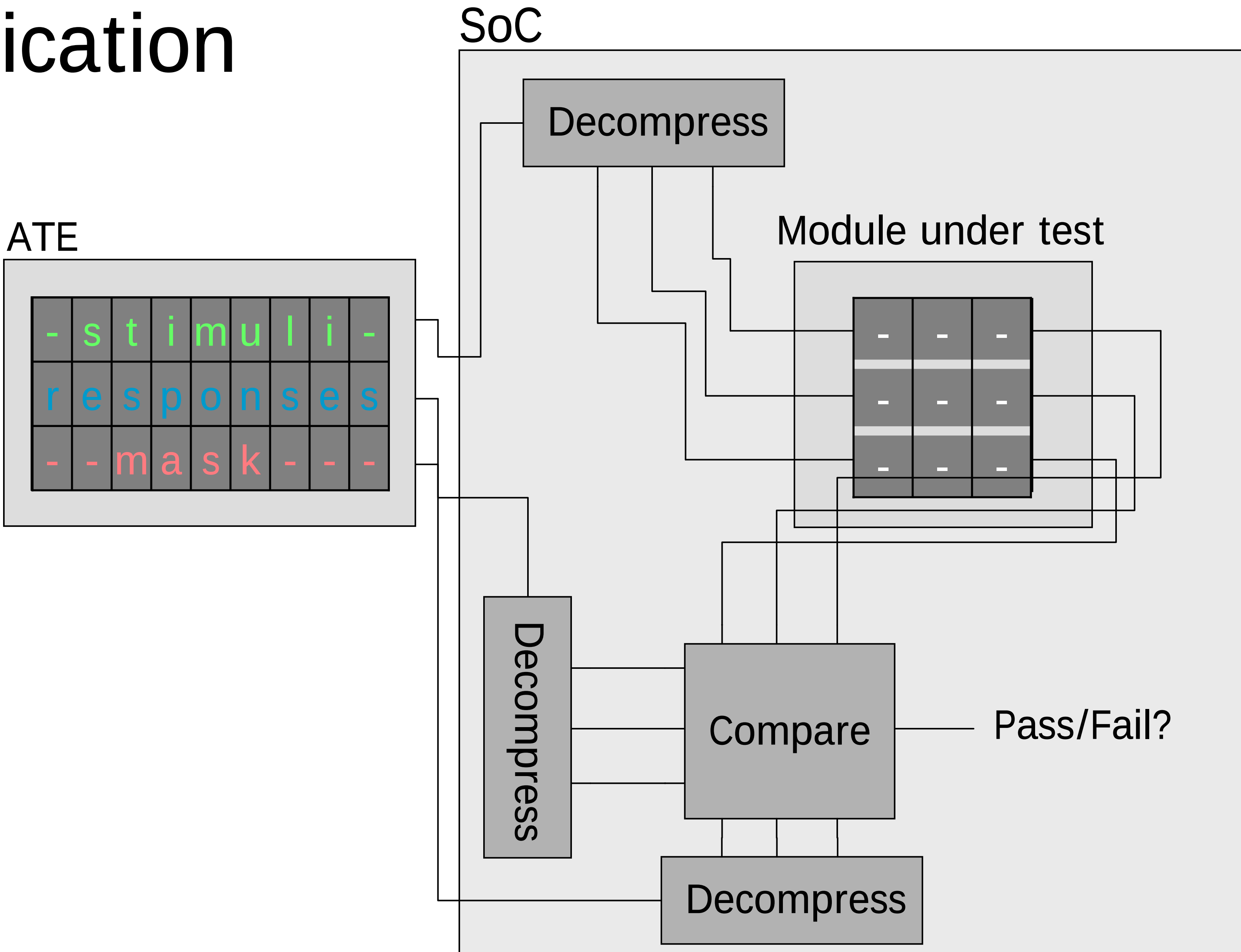
Approach

For all modules

- Define Mask
- Fill all don’t care bits
- Compress test stimuli, expected responses, and mask data
- **Design de-compression logic**

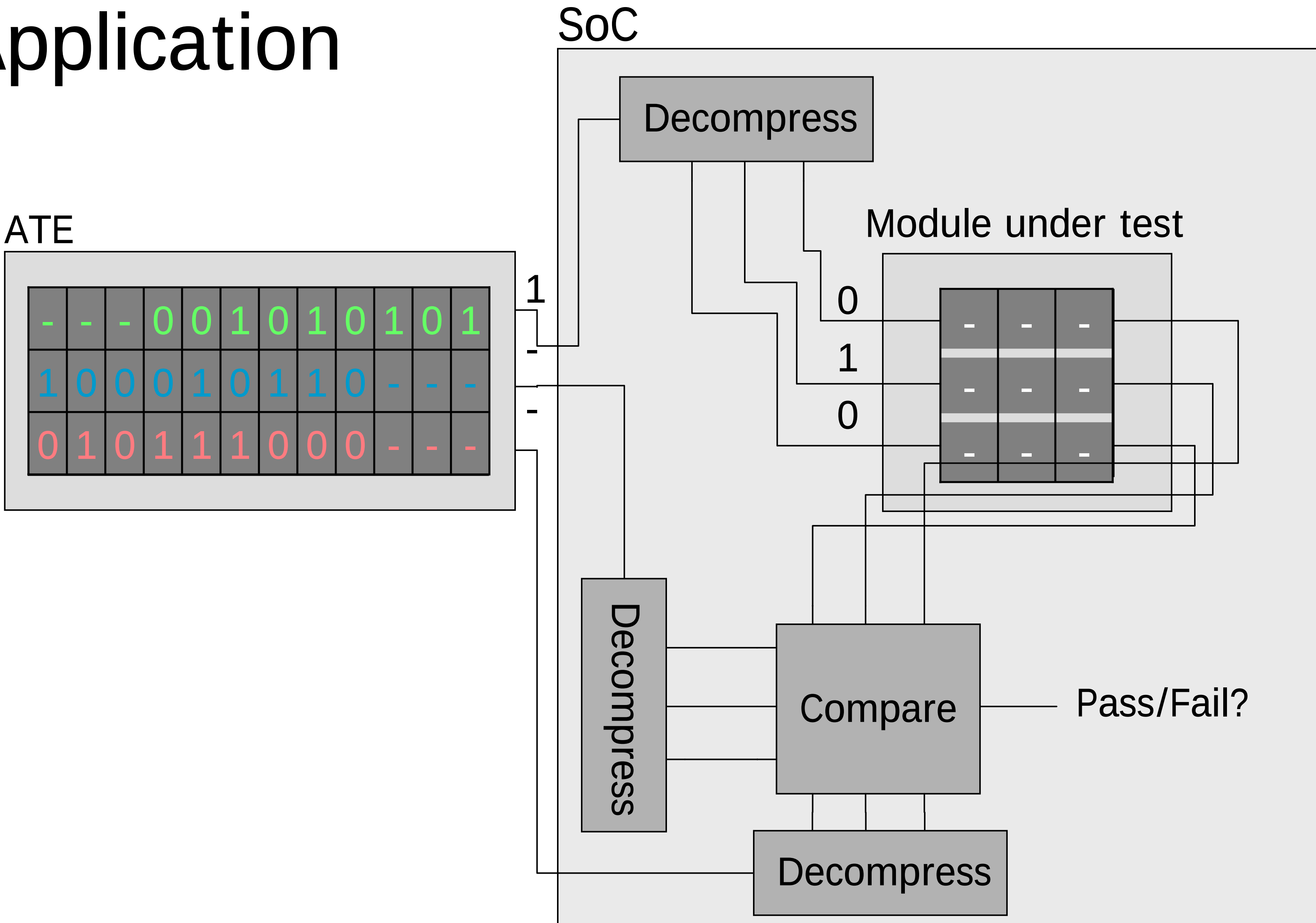
3. Test Architecture

Application



-	s	t	i	m	u	l	i	-
r	e	s	p	o	n	s	e	s
-	-	m	a	s	k	-	-	-

Application

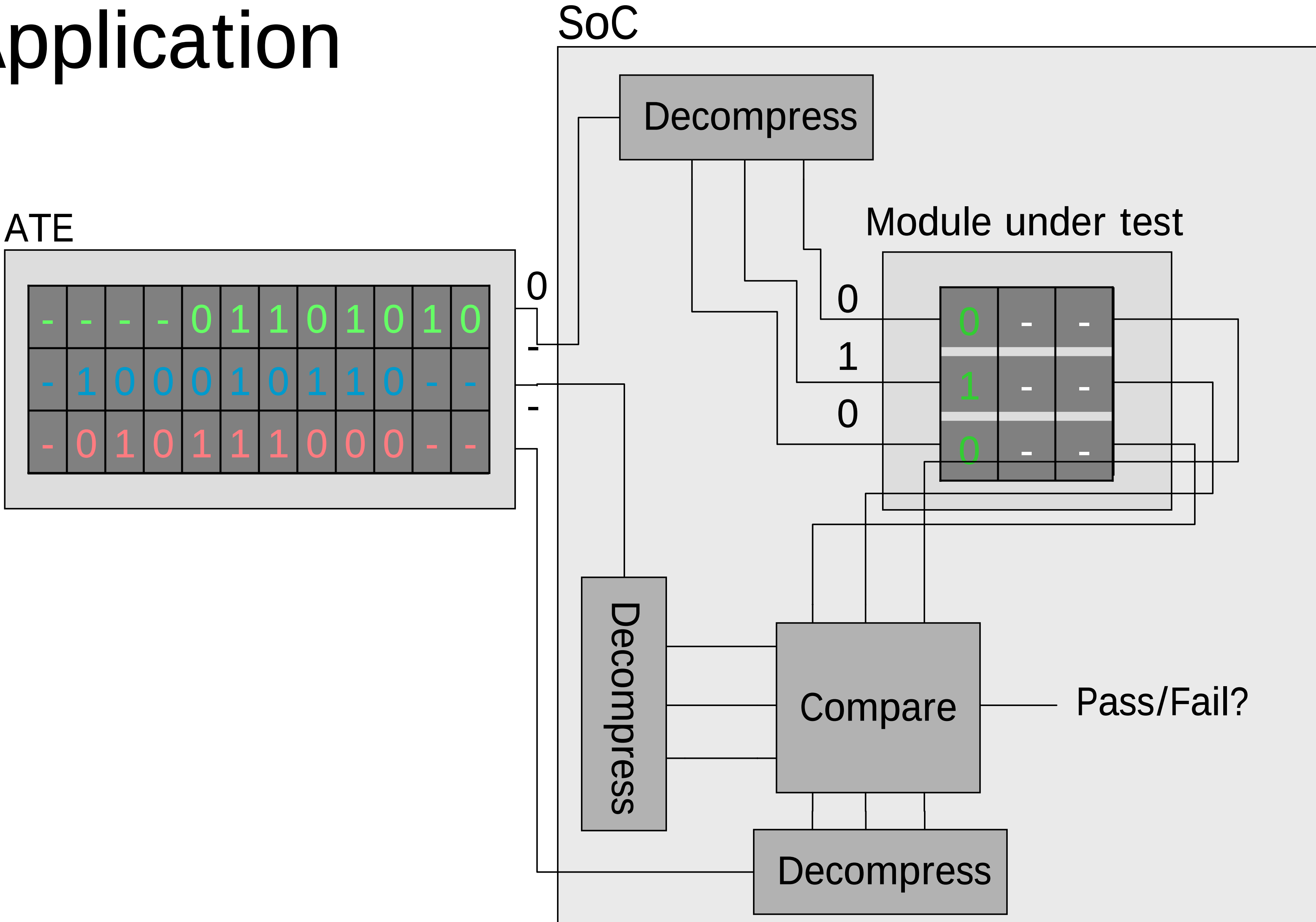


-	-	-	0	0	1	0	1	0	1	0	1
1	0	0	0	1	0	1	1	0	-	-	-
0	1	0	1	1	1	0	0	0	-	-	-

-	-	-
-	-	-
-	-	-

3. Test Architecture

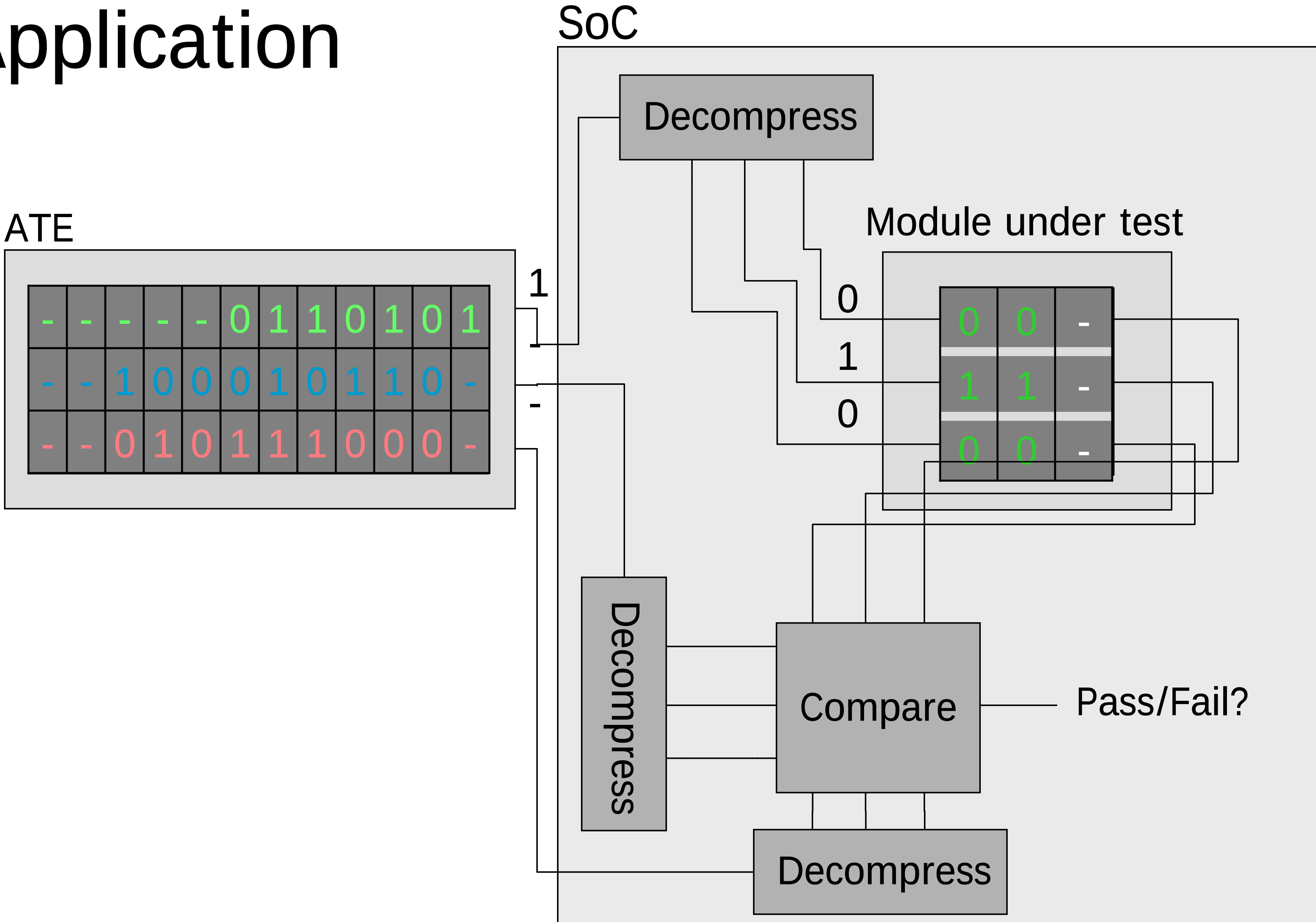
Application



-	-	-	-	0	1	1	0	1	0	1	0
-	1	0	0	0	1	0	1	1	0	-	-
-	0	1	0	1	1	1	0	0	0	-	-

0	-	-
1	-	-
0	-	-

Application

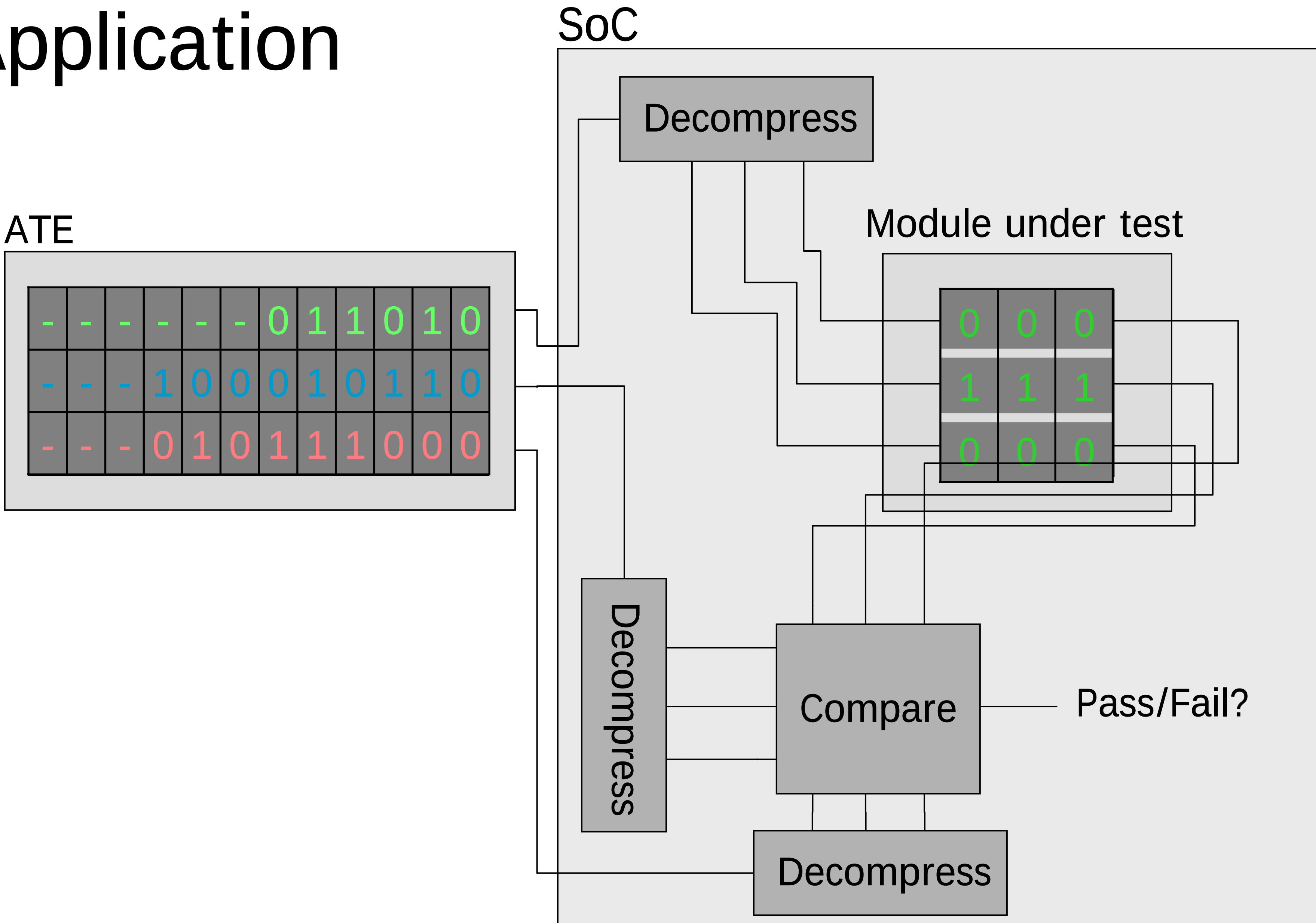


-	-	-	-	-	0	1	1	0	1	0	1
-	-	1	0	0	0	1	0	1	1	0	-
-	-	0	1	0	1	1	1	0	0	0	-

0	0	-
1	1	-
0	0	-

3. Test Architecture

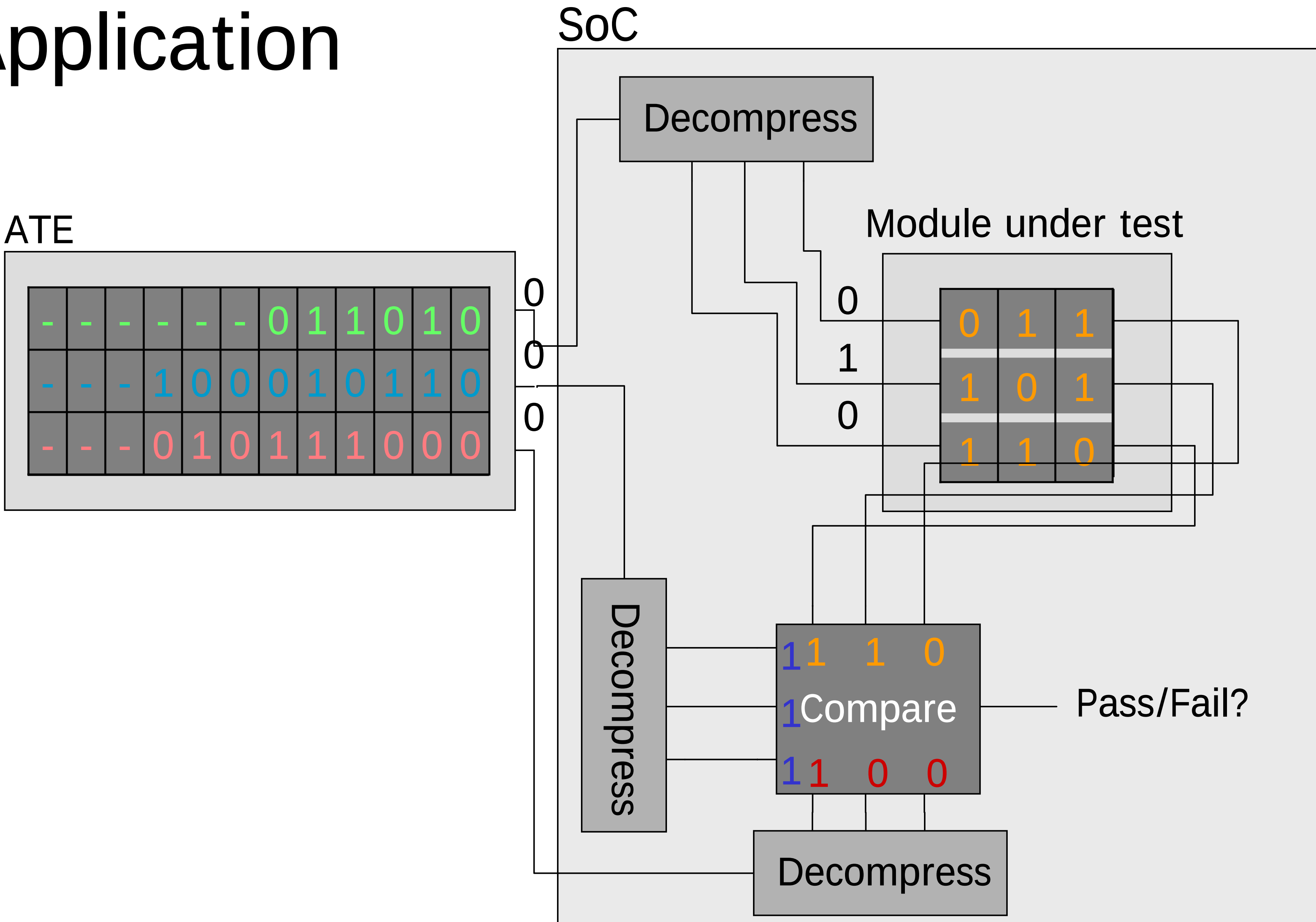
Application



-	-	-	-	-	-	0	1	1	0	1	0
-	-	-	1	0	0	0	1	0	1	1	0
-	-	-	0	1	0	1	1	1	0	0	0

0	0	0
1	1	1
0	0	0

Application



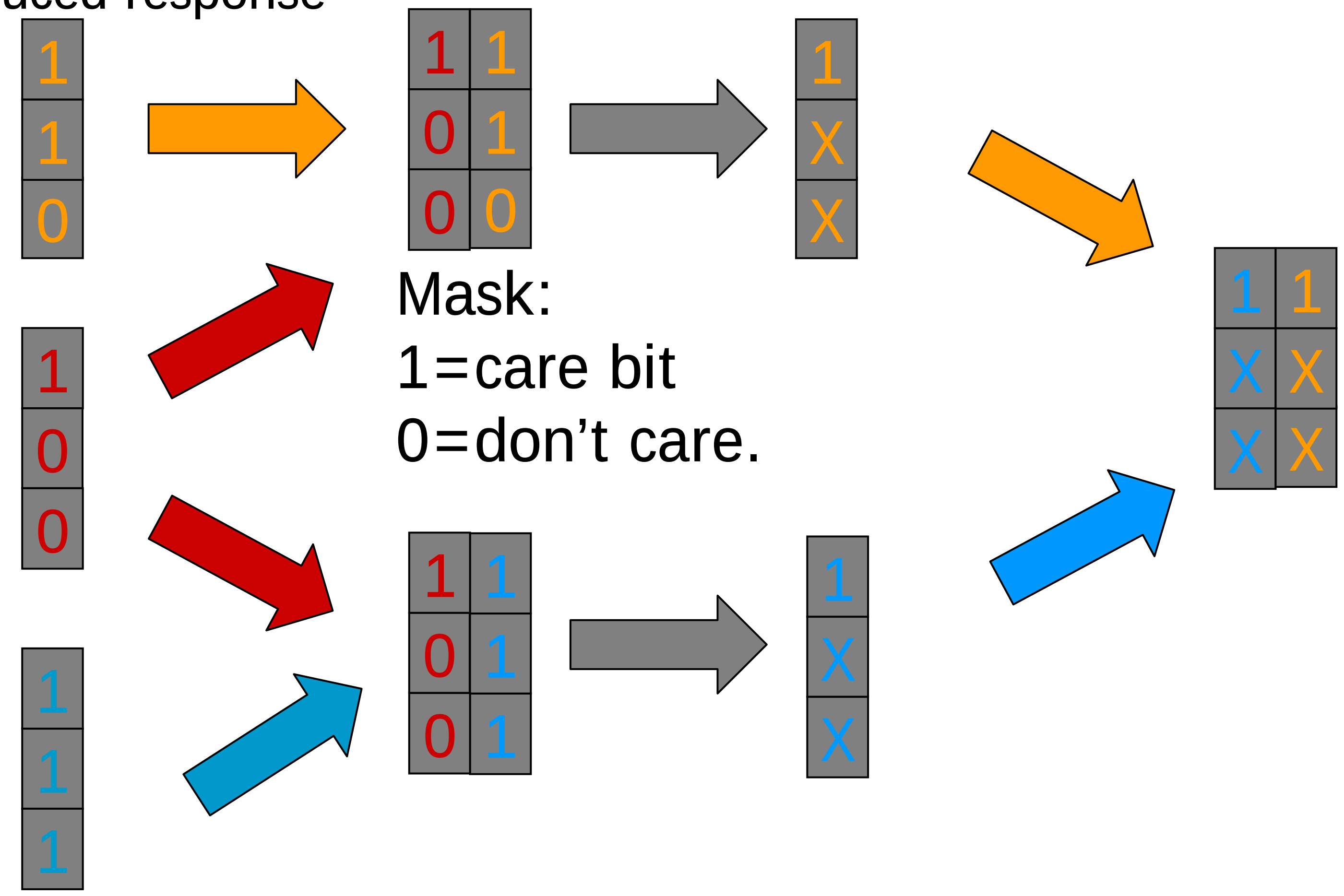
-	-	-	-	-	-	0	1	1	0	1	0
-	-	-	1	0	0	0	1	0	1	1	0
-	-	-	0	1	0	1	1	1	0	0	0

0	1	1
1	0	1
1	1	0

3. Test Architecture

Application

produced response



mask

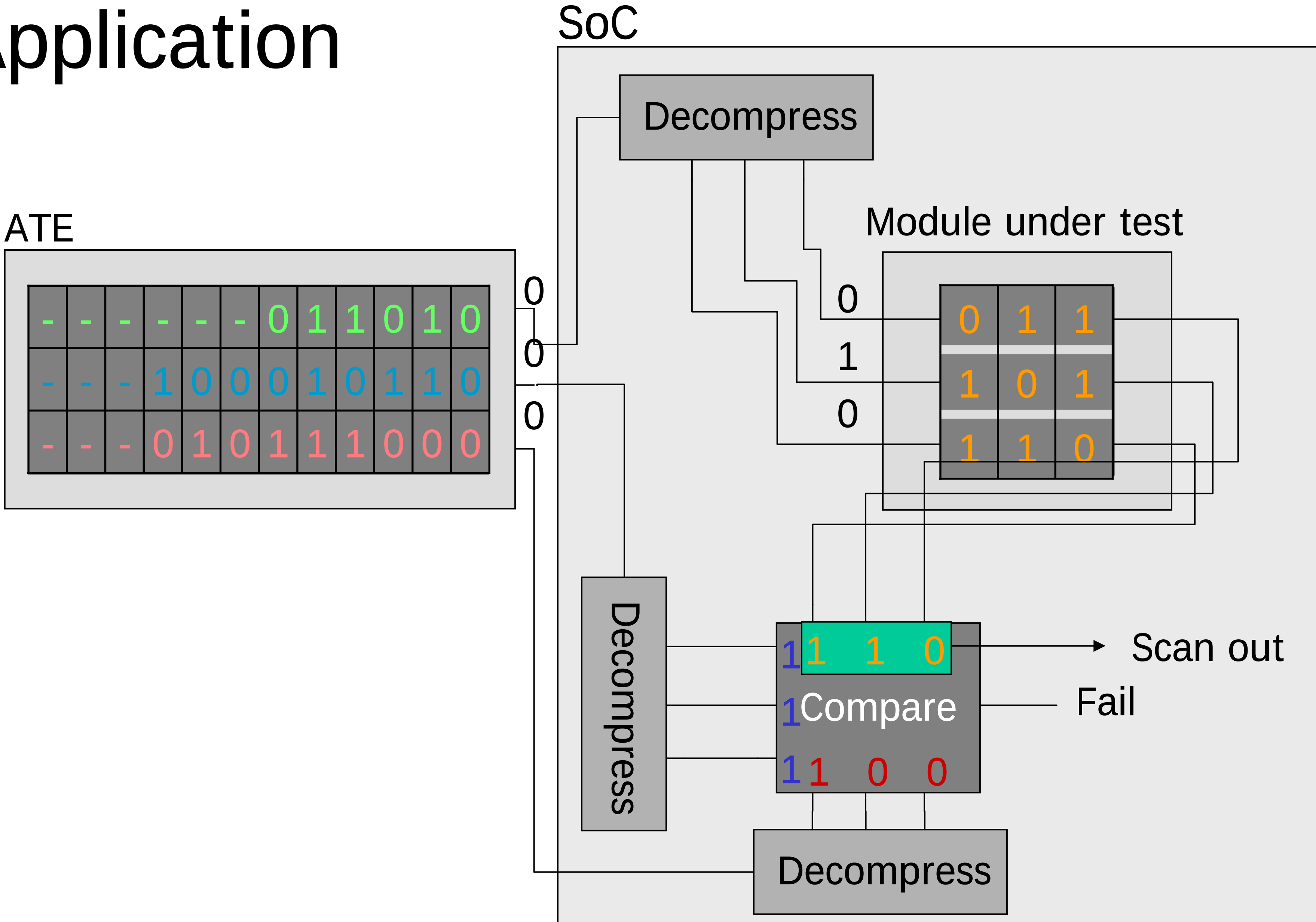
Mask:
1=care bit
0=don’t care.

Pass

expected (unmasked) response

3. Test Architecture

Application



ATE

-	-	-	-	-	-	0	1	1	0	1	0
-	-	-	1	0	0	0	1	0	1	1	0
-	-	-	0	1	0	1	1	1	0	0	0

Module under test

0	1	1
1	0	1
1	1	0

Outline

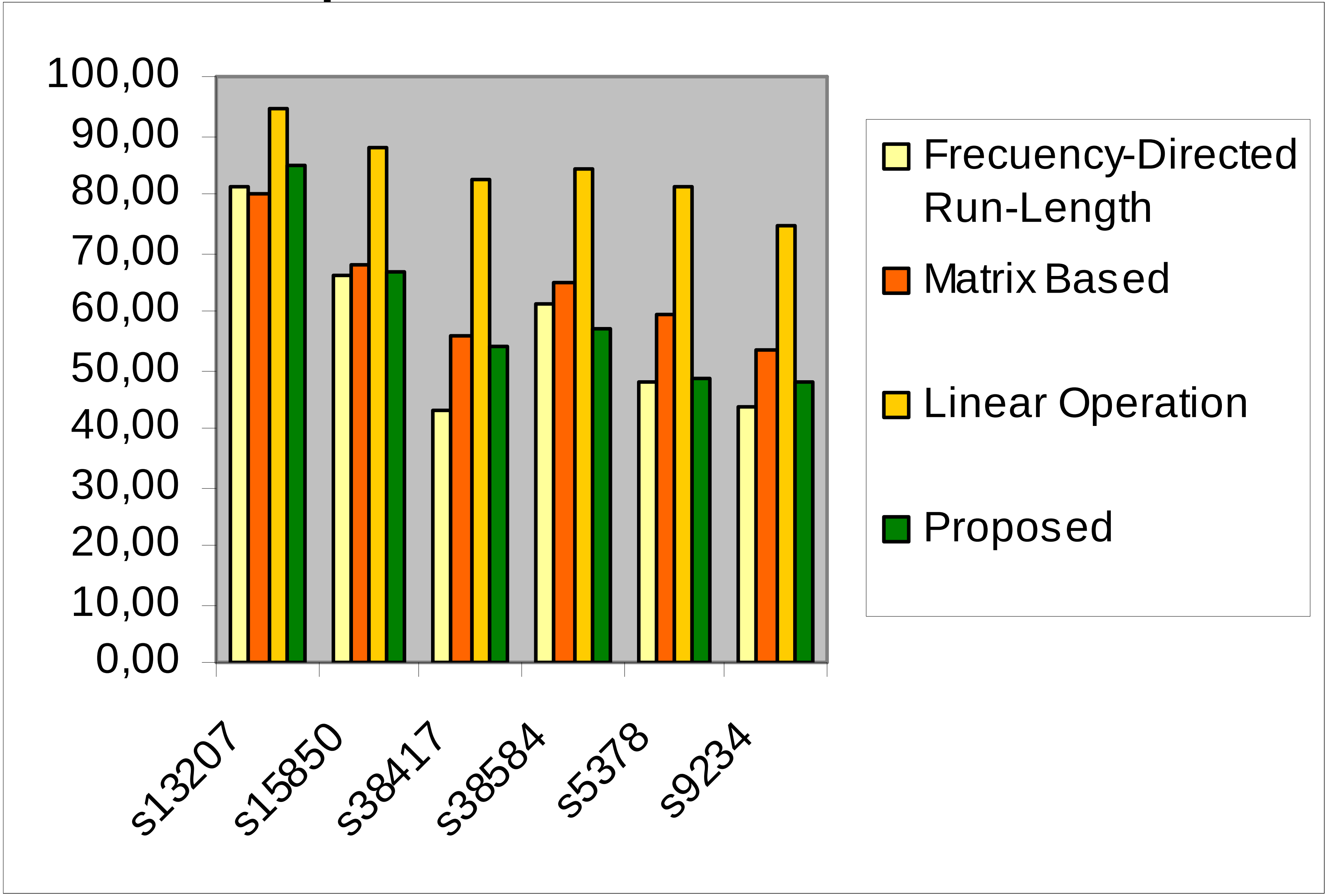
1. Introduction
2. Prior Work
3. Test Architecture
4. Experimental Results
5. Conclusion

Experimental Setup

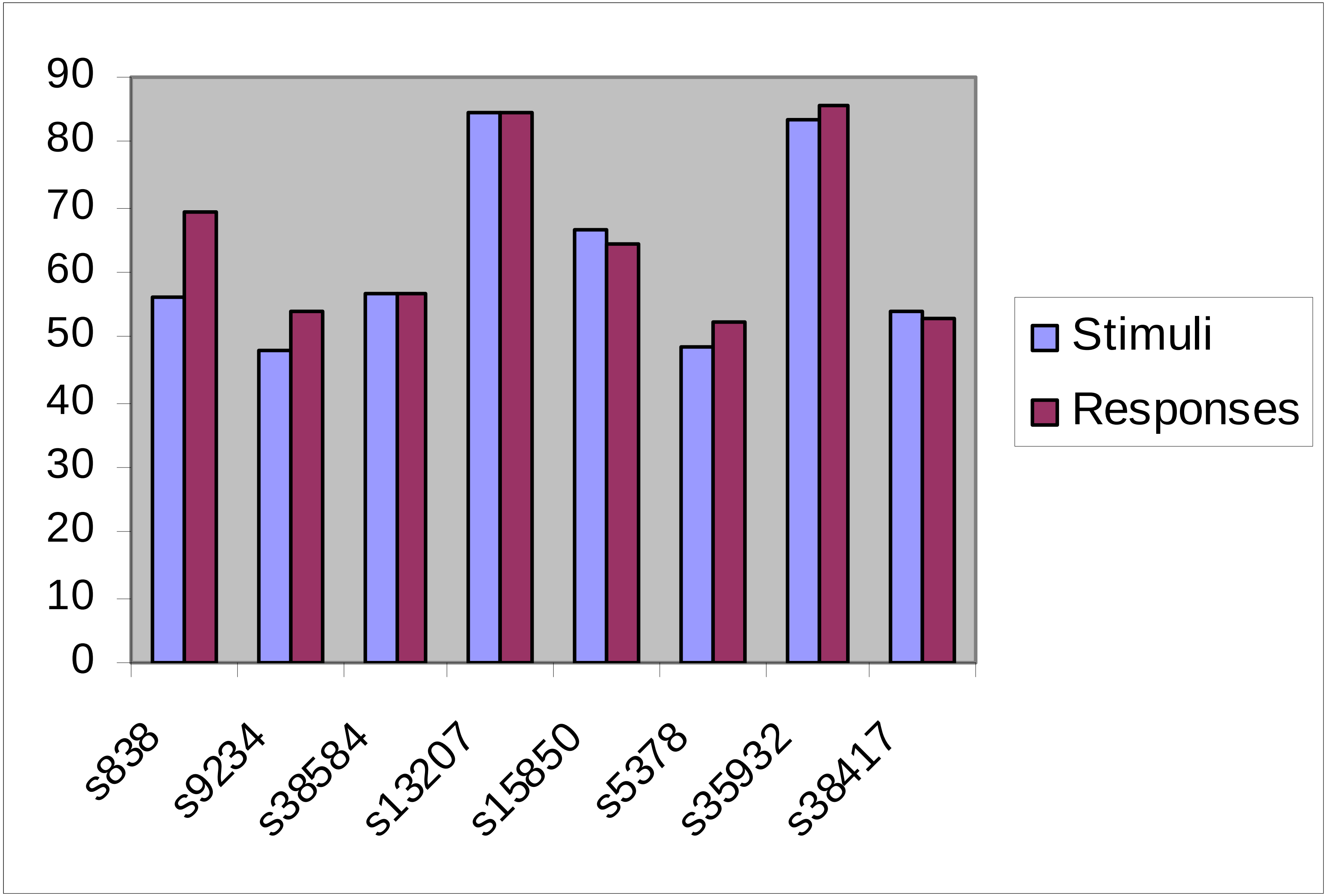
- ISCAS modules and an industrial design
- Facsmile coding; small decompression test program (88 bytes)
- Processor (on-chip or off-chip) for decompression and test evaluation
- Compression=(Original bits – Compressed bits)/Original bits

4. Experimental Results

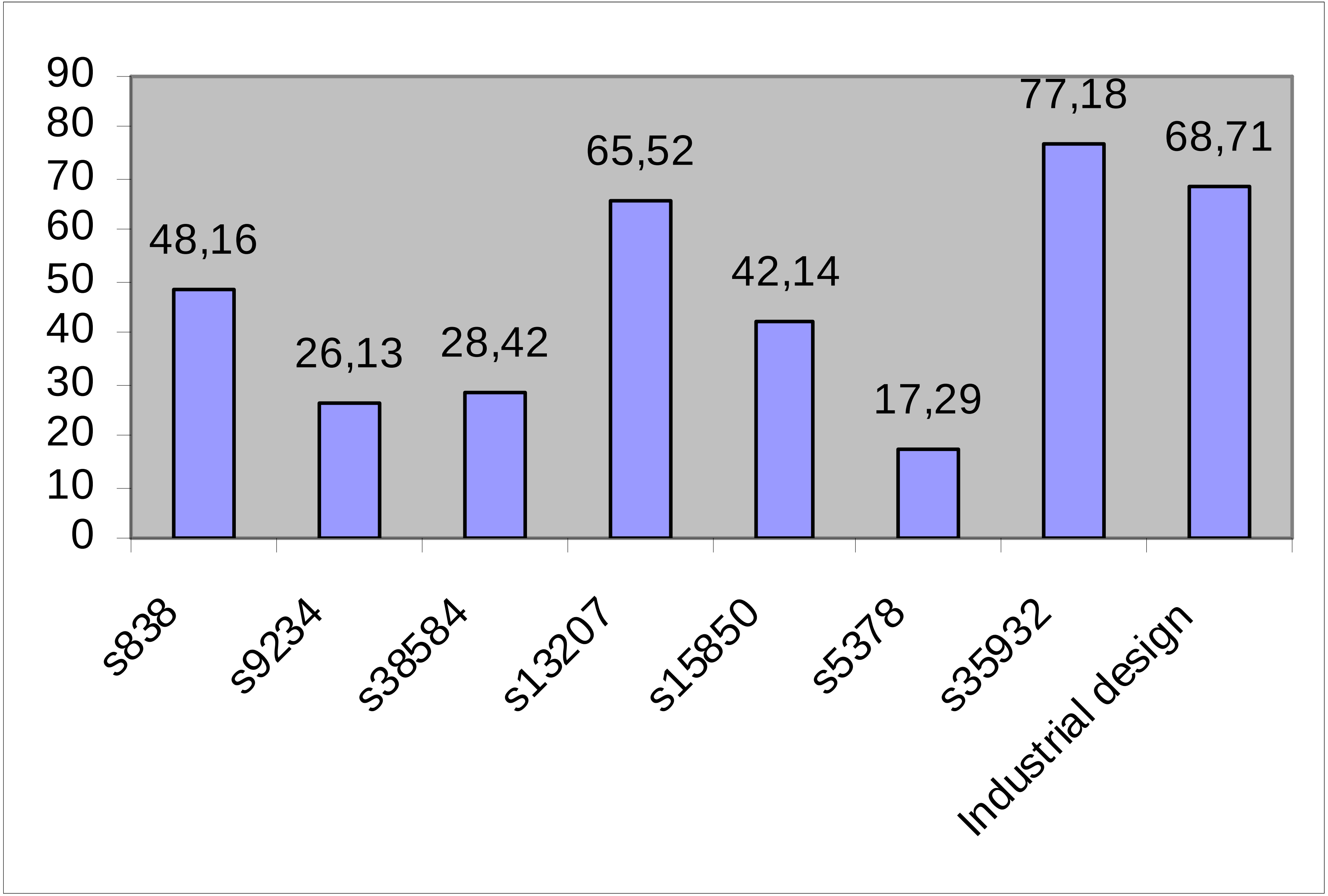
Stimuli compression



Stimuli and responses compression

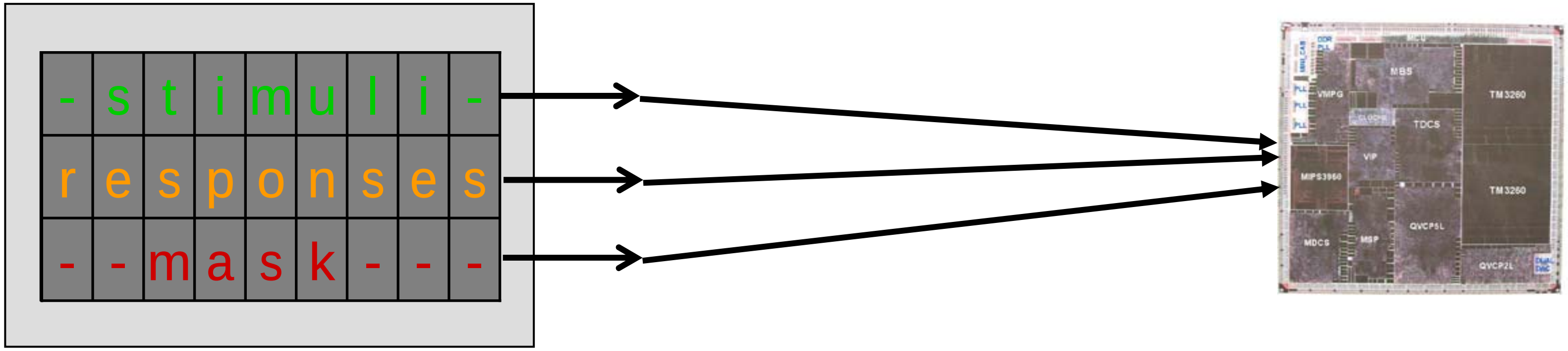


Stimuli, responses and mask compression



4. Experimental Results

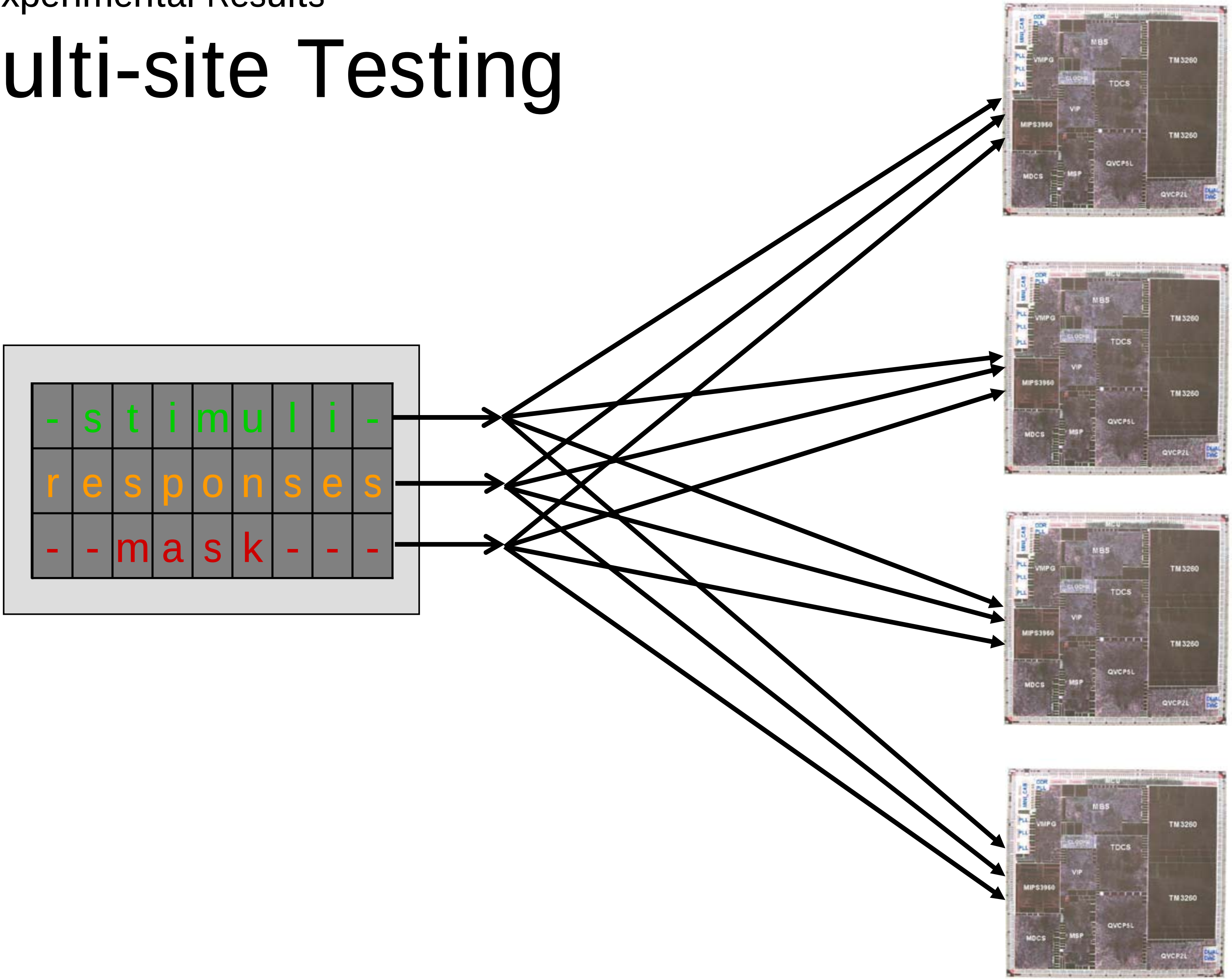
Multi-site Testing



-	s	t	i	m	u	l	i	-
r	e	s	p	o	n	s	e	s
-	-	m	a	s	k	-	-	-

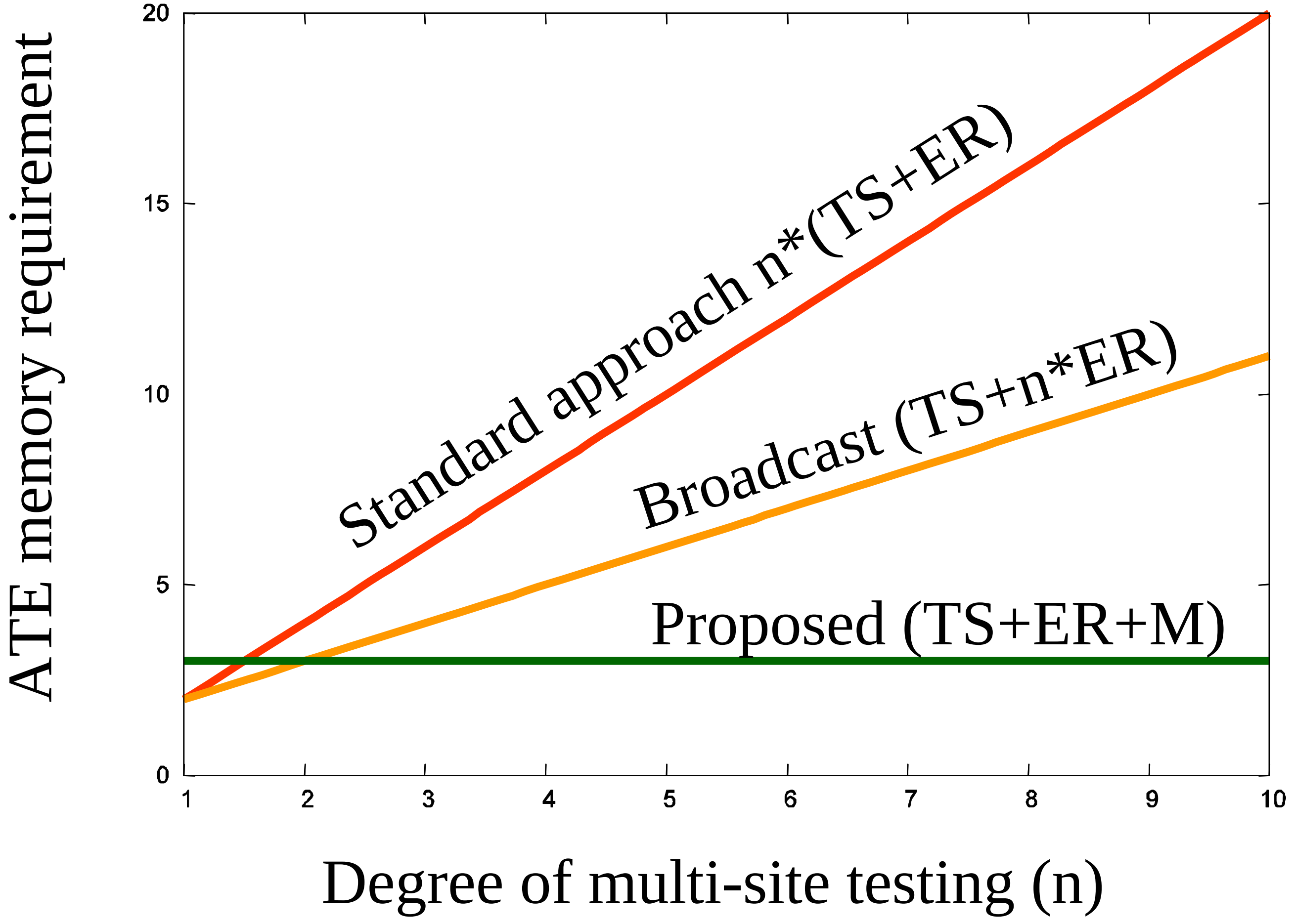
4. Experimental Results

Multi-site Testing



4. Experimental Results

ATE Memory Usage



Conclusions

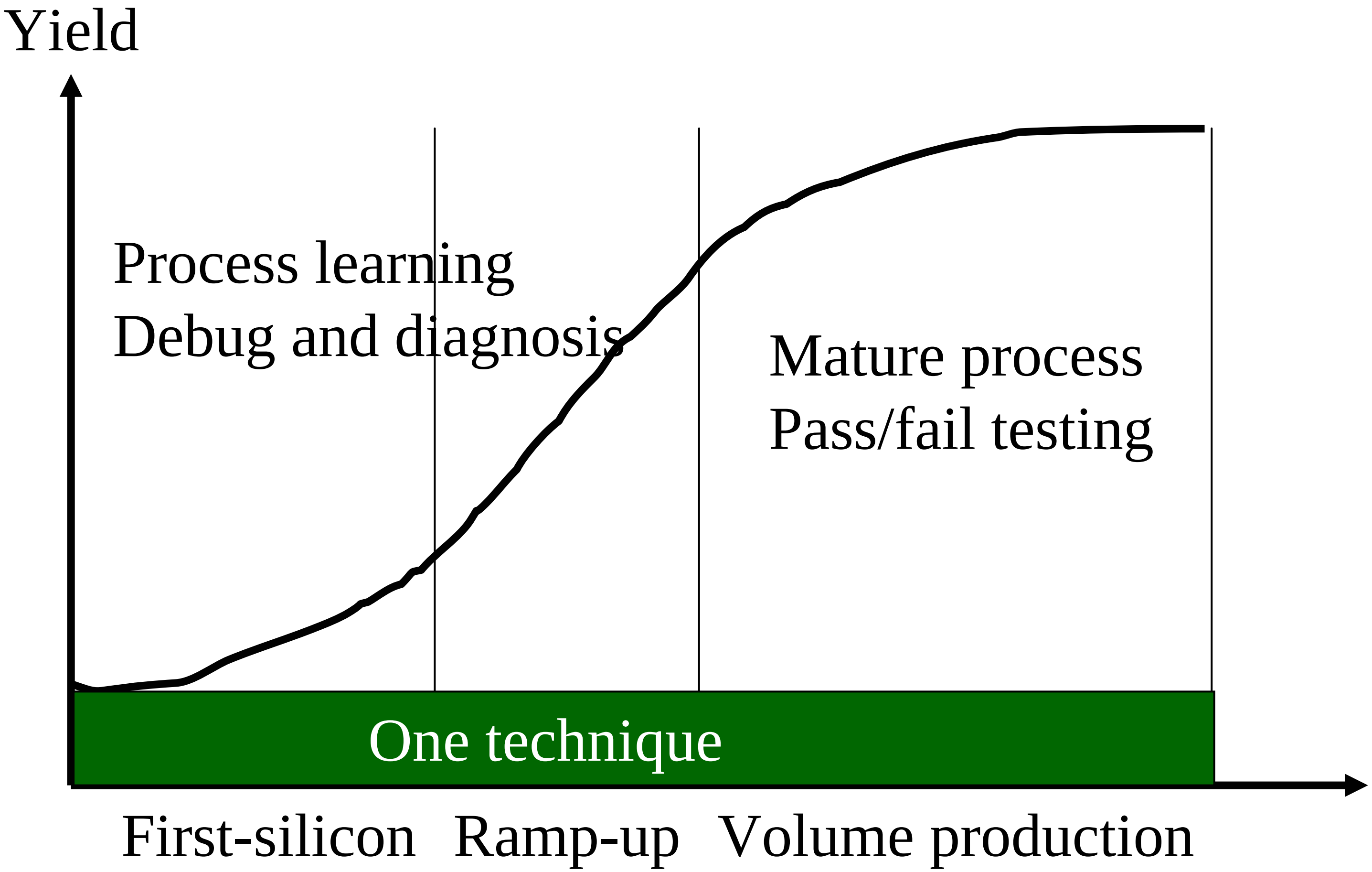
Problems when testing ICs:

- Long test times
- High ATE memory requirement
- Low throughput

- Abort-on-Fail testing can reduce the test application times
- Test data compression can due reduce the ATE memory need
- Multi-site test requires ATE memory

- Defined: A SoC Test Compression Architecture for abort-on-fail testing and ATE memory minimization
  – Debug and diagnosis
  – Test data compression
  – Pass/fail test - Abort-on-fail testing (clock cycle granularity)
  – 100% X-tolerant
  – Constant ATE memory requirement at multi-site test

Conclusions



Yield

Process learning

Debug and diagnosis

Mature process

Pass/fail testing

One technique

First-silicon Ramp-up Volume production