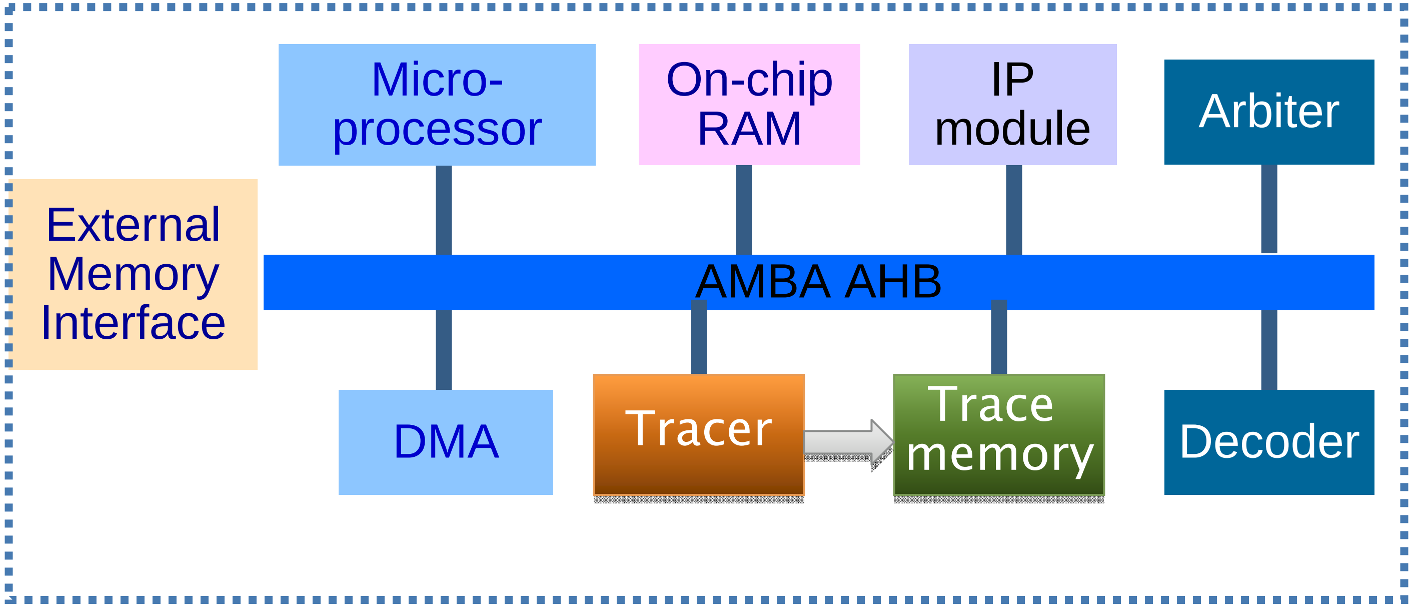
A Reverse-Encoding-based on-chip AHB Bus Tracer for Efficient Circular Buffer Utilization

Presenter : Fu-Ching Yang

Fu-Ching Yang, Cheng-Lung Chiang, and Ing-Jer Huang
Department of Computer Science and Engineering
National Sun Yat-Sen University

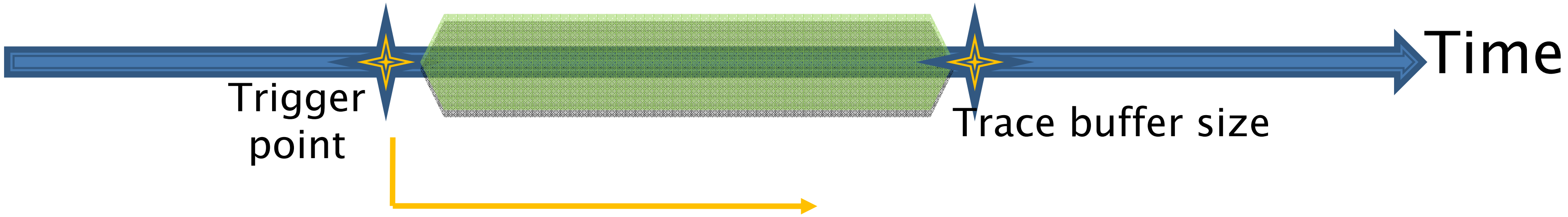
Tracing is important in SoC monitoring and debugging

- Debugging in SoC
  - To embed the hardware tracer to capture and compress the signals in realtime
  - Store the data in the on-chip trace memory (circular buffer)



Pre-T/Post-T traces are important in SoC monitoring and debugging

- There are two types of trace
  - Post-Triggering (Post-T) trace



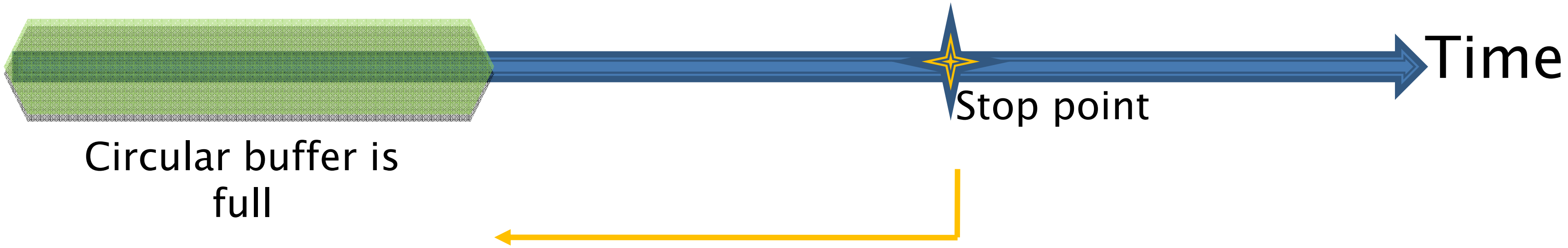
Time

Trigger point

Trace buffer size

Trace after the start point

  - Pre-Triggering (Pre-T) trace



Time

Stop point

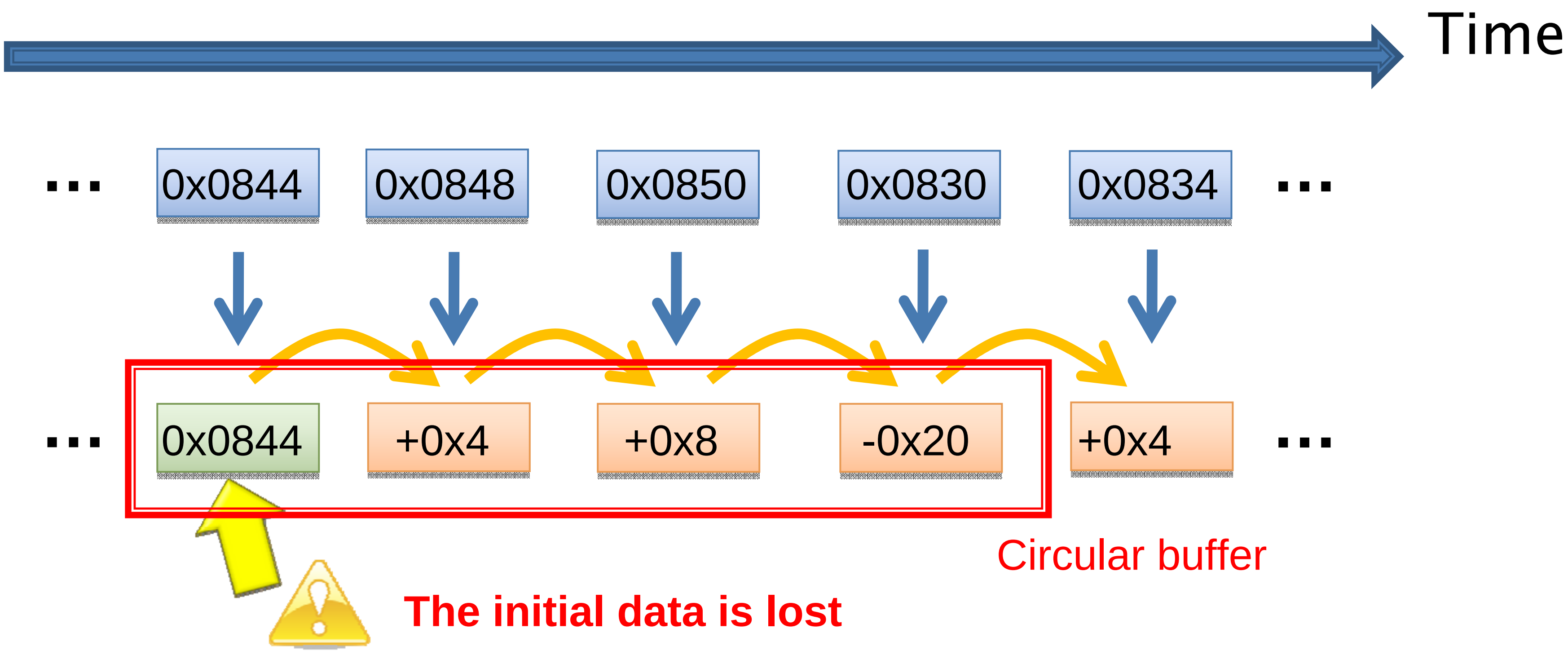
Circular buffer is full

Trace before the stop point

Problems with Pre-T trace in the forward encoding compression method

- Forward encoding differential compression

🟦 : Input Data 🟩 : Initial data 🟧 : Encoded data



Time

… 0x0844 0x0848 0x0850 0x0830 0x0834 …

… 0x0844 +0x4 +0x8 -0x20 +0x4 …

Circular buffer

The initial data is lost

- Traditional approaches (called *Forward encoding*) do not support compression onto a circular buffer well

Outline

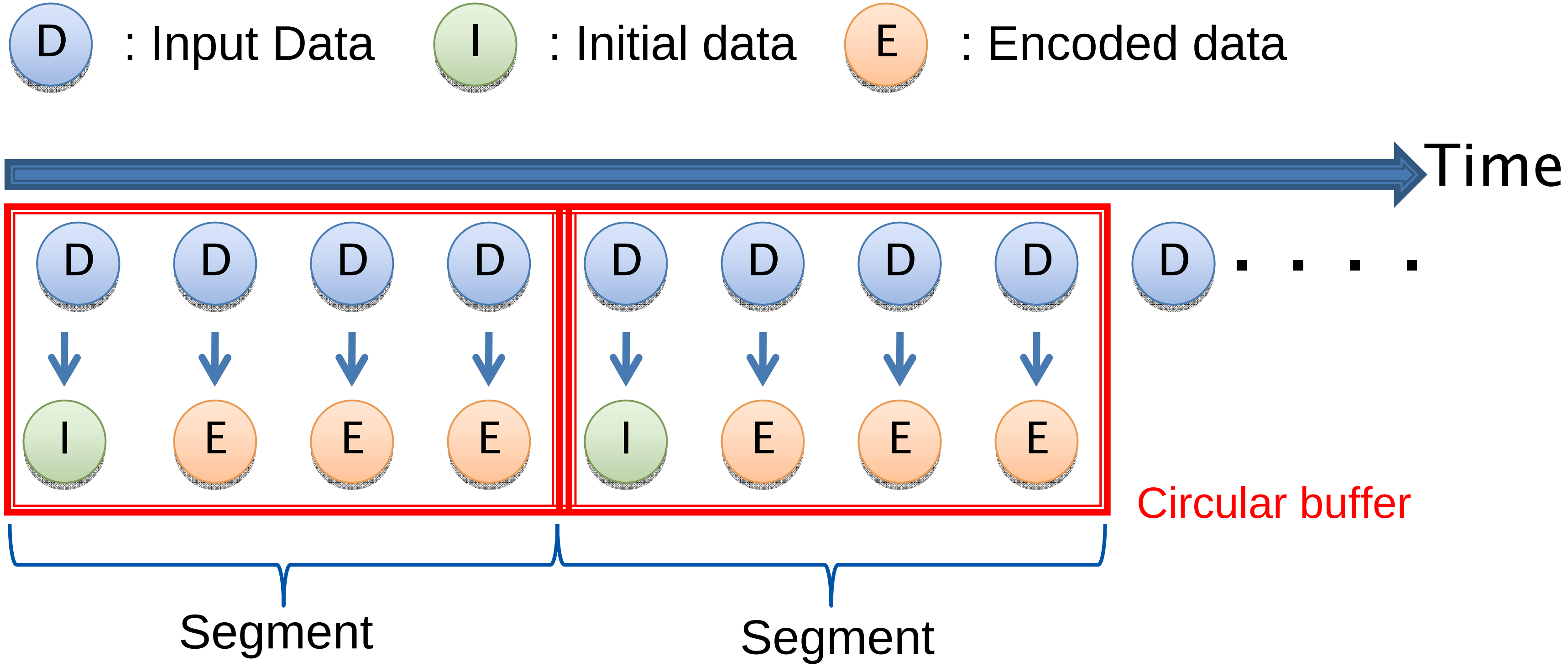
- Related work
  - Periodical triggering
- Proposed reverse encoding
  - Concept
  - Applying on the differential and slice compression
  - Applying on the dictionary-based compression
- Experiment
- Summary

Related work – Pre-T tracers

1. Pre-T trace, no compression
   - Conventional logic analyzer
   - LEON3 AHBTRACE
2. Pre-T trace with compression, but low compression ratio
   - Compression based on data filtering
     - Branch/target filtering
       - *Mann, ARM ETM, and NEXUS interface*
     - Run-length encoding
       - *Some logic analyzers*
   - Low compression ratio
3. Pre-T trace, with good compression ratio
   - Lin et al. propose periodical triggering [6]

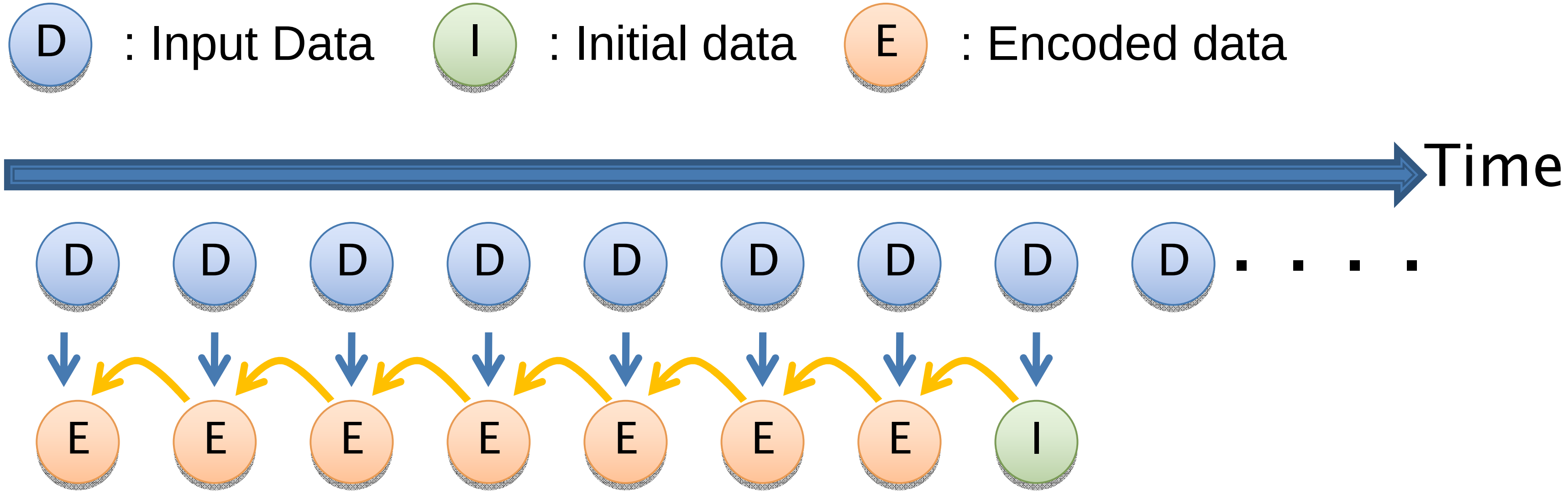
Related work: Periodical Triggering

- Periodical triggering concept
  - Divide the trace into small segments
  - Isolate the encoding relationship between segments



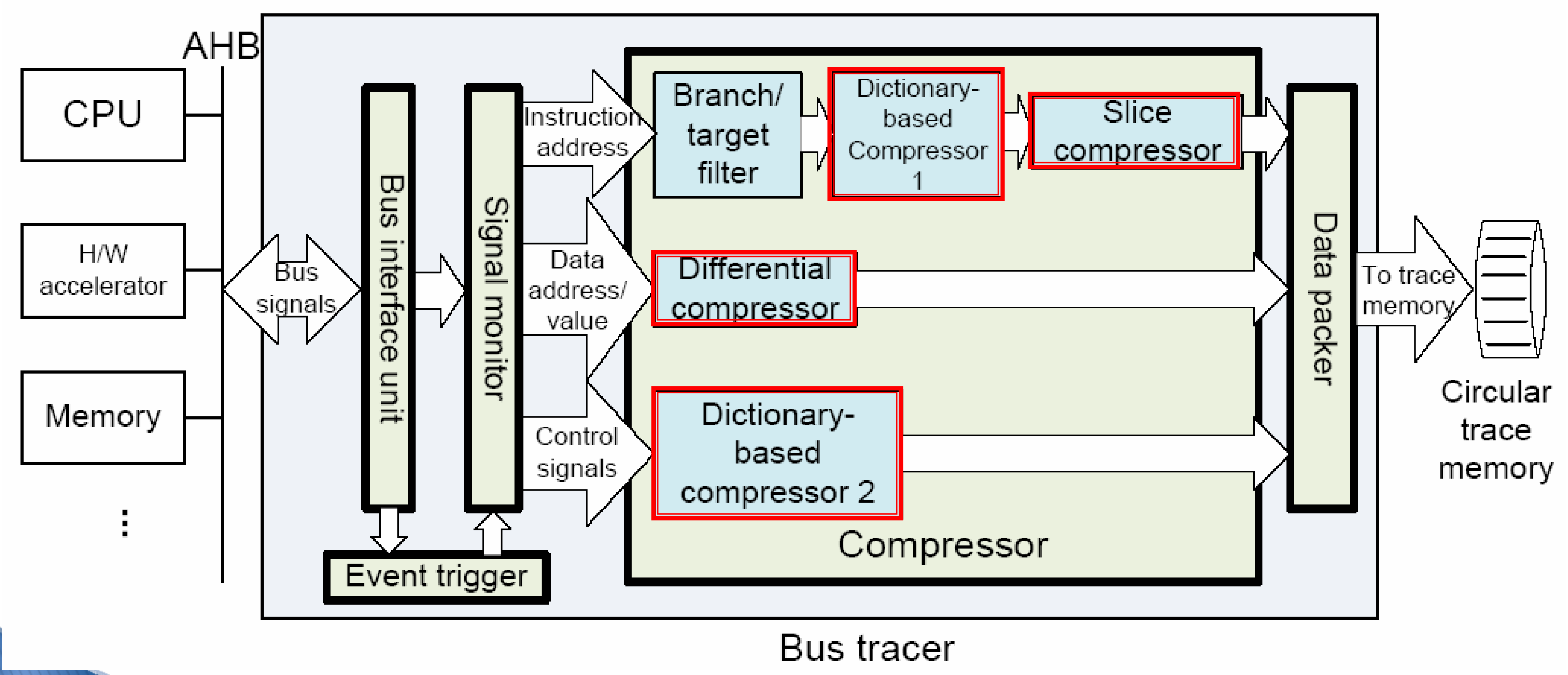
Proposed reverse encoding

- Record the uncompressed data after the referenced, encoded data
- It can be applied to any compression algorithm encoded based on data relationship



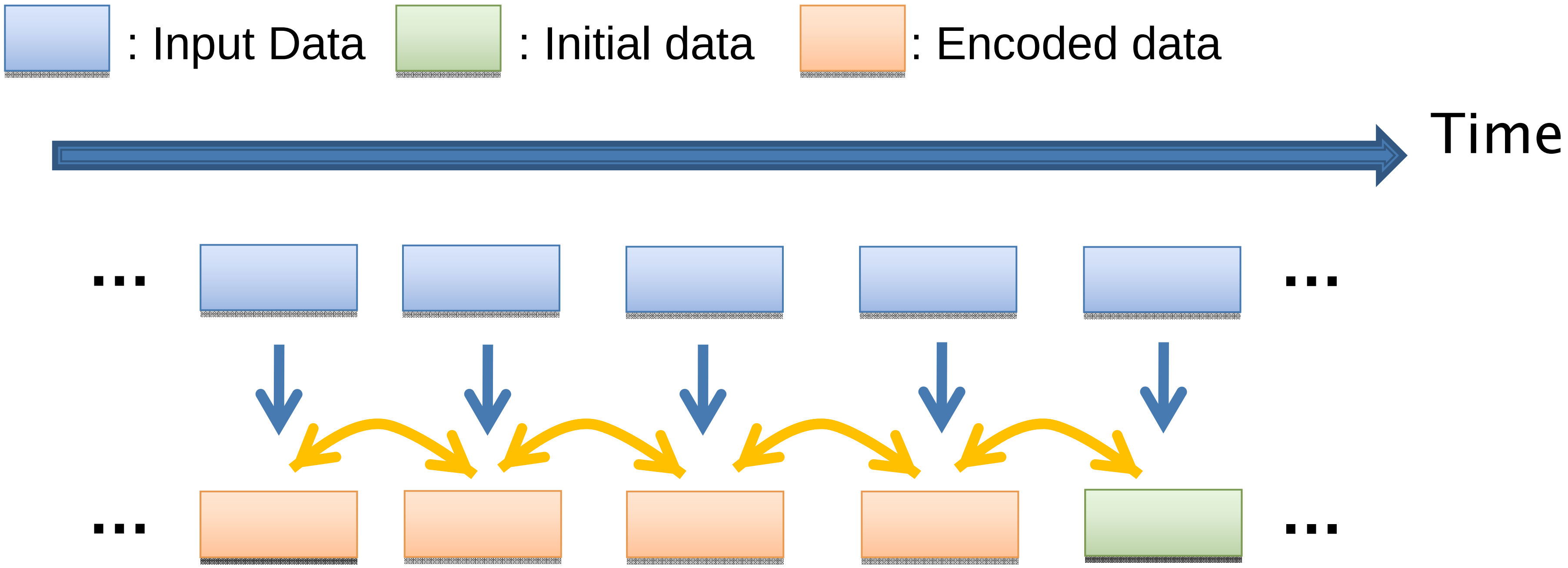
The proposed bus tracer supporting both Pre-T/Post-T trace

- Apply the reverse encoding algorithm on the forward encoding tracer
  - Differential compression
  - Slice compression
  - Dictionary-based compression



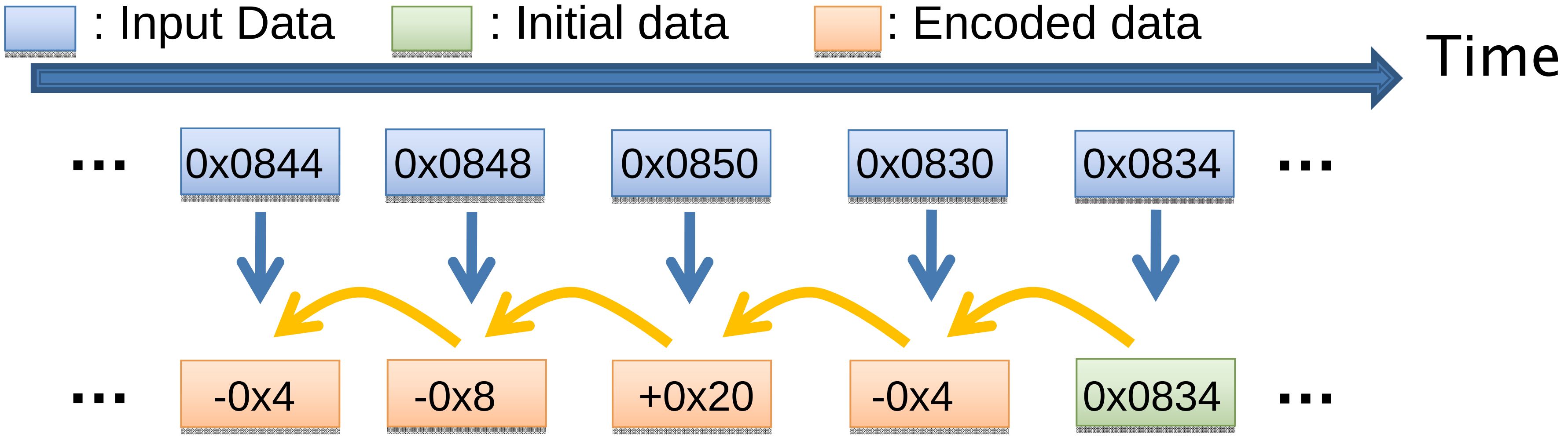
Applying reverse encoding

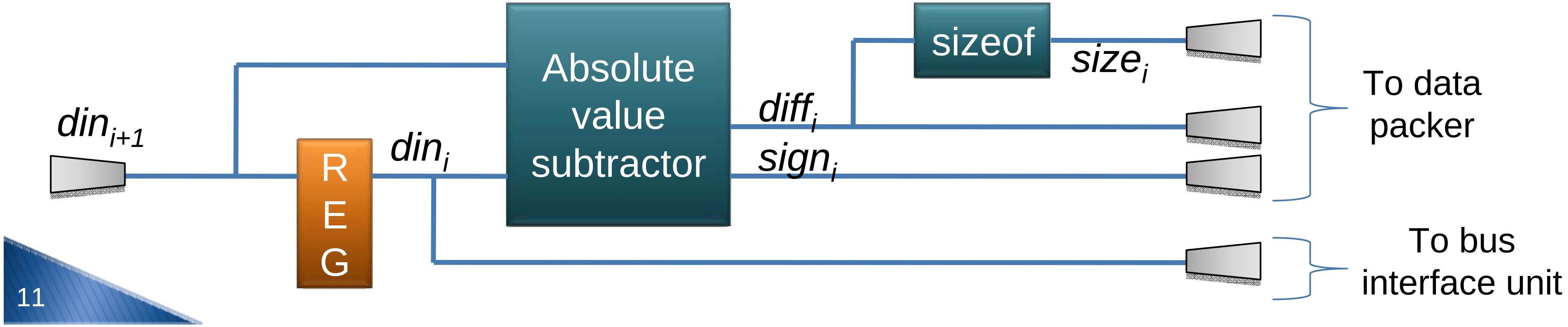
- Related data are *adjacent*
  - Differential compression
  - Slice compression



Reverse encoding based differential compression

- Switch the two input
- Output the last data in uncompressed format
- Significant small hardware overhead





Reverse encoding based slice compression

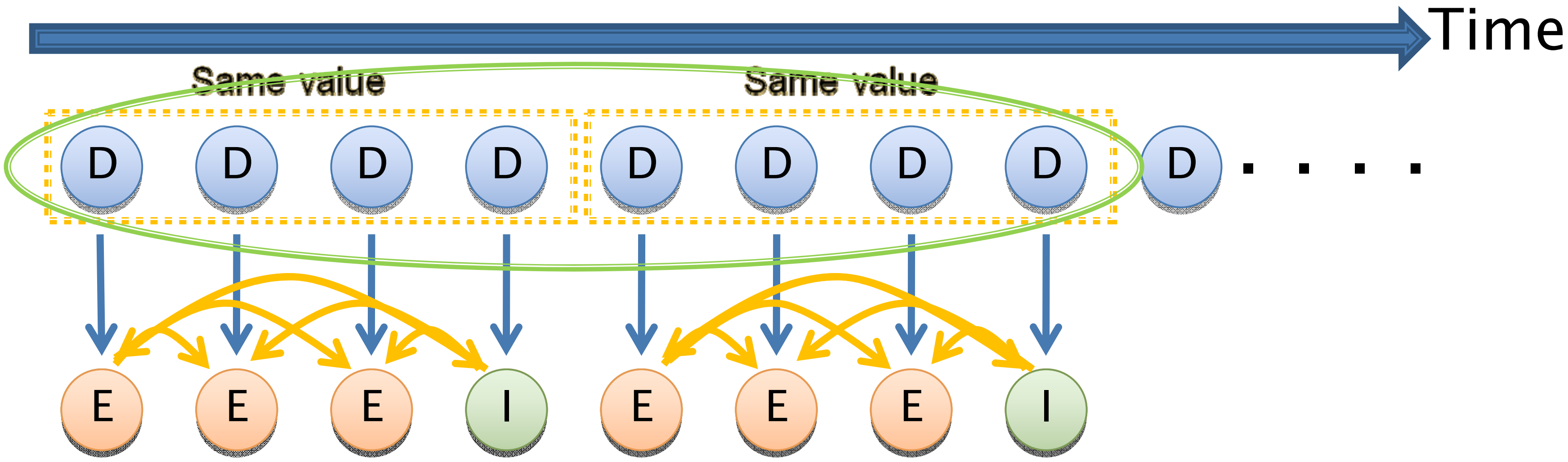
Time	Original address	Forward encoding	Reverse encoding
t	0x0300_6600	0x0300_6600	0x600
t+1	0x0300_6120	0x120	0x20
t+2	0x0300_6130	0x30	0x300_6130
t+3	0x0080_4020	0x080_4020	0x020
t+4	0x0080_4400	0x400	0x0080_4400

- Switch the two input
- Output the last data in uncompressed format
- Significant small hardware overhead

Forward encoding dictionary-based approach

- Related data are *not adjacent*
  - Dictionary-based compression
    - LZ compression

D : Input Data I : Initial data E : Encoded data

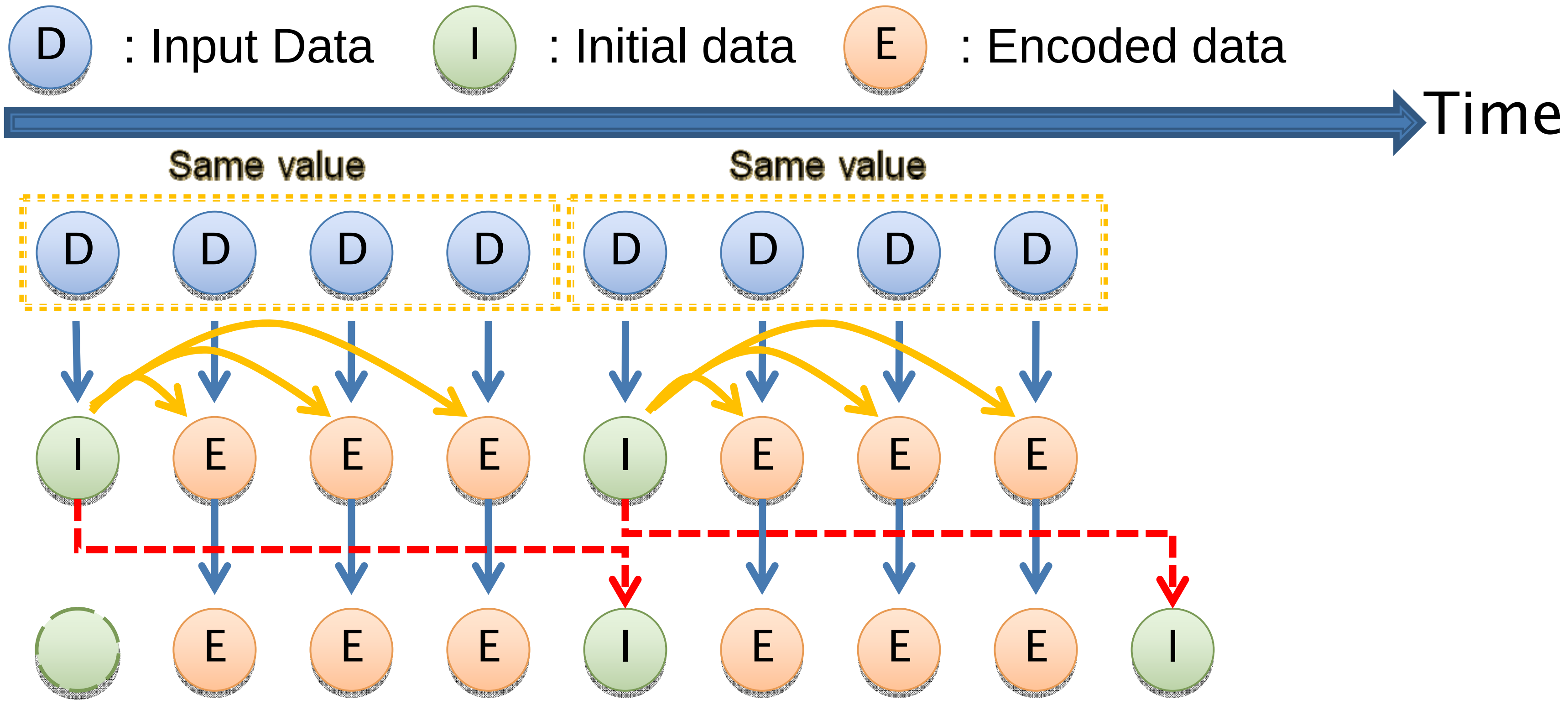


Assume dictionary table size: 1

⚠ Impractical hardware design

Applying reverse encoding

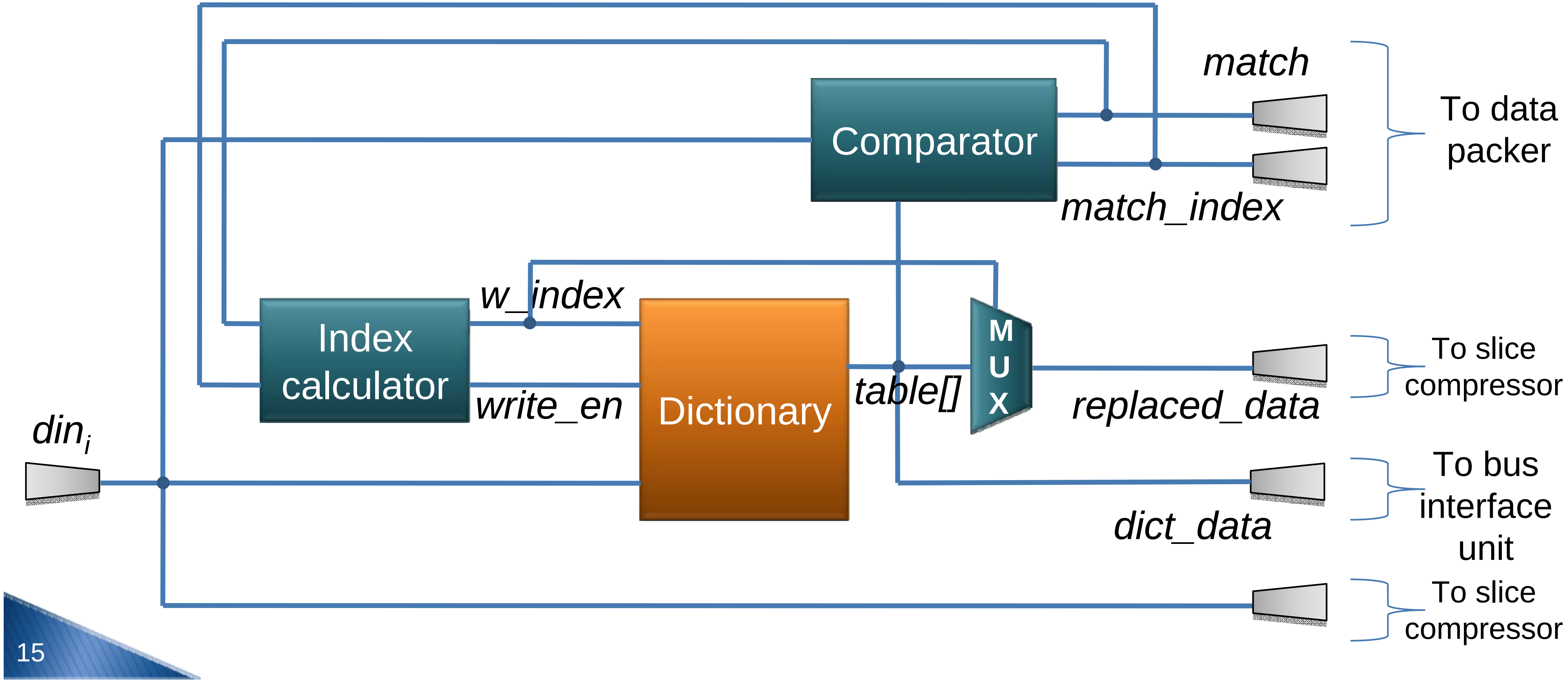
- *Delay the output* of the Miss data
  - The Miss data output only if they are replaced
  - The initial table is output
    - The decreased compression ratio can be ignored



Assume dictionary table size: 1

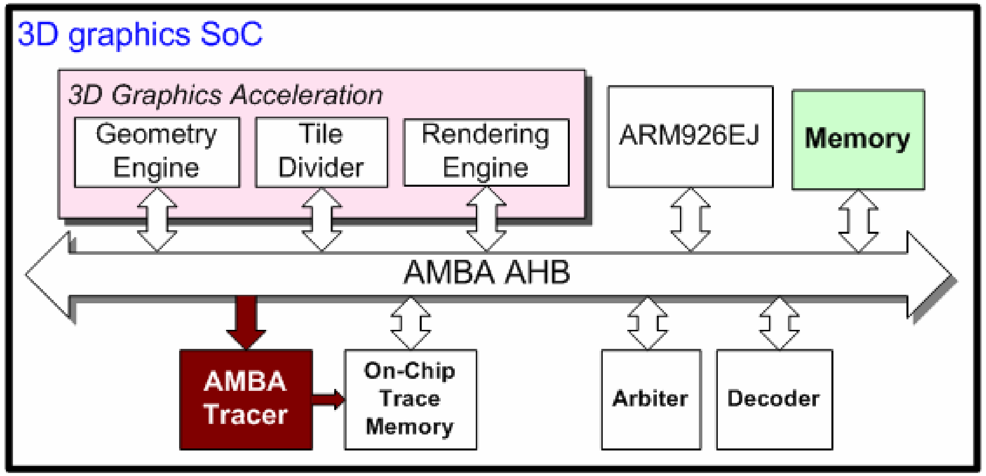
Reverse encoding based Dictionary-based compression

- The Miss data output only if they are replaced
- Output the dictionary table at the end
- Significant small hardware overhead



Experiment

- A reverse-encoding-based on-chip AHB bus tracer is implemented
  - Support both Pre-T and Post-T trace
- 5 C-based benchmarks
  - Loop and recursive intensive
- Experiment environment: ARM EASY



Compression ratio of forward encoding and reverse encoding

- The compression ratio is the same to the forward encoding
  - Reverse encoding only re-arrange the order
  - The compression ratio of the dictionary-based compression drops slightly
    - The dictionary table size is relatively smaller than the total trace size

Benchmark	Instruction address		Data value	
	Forward encoding	Reverse encoding	Forward encoding	Reverse encoding
Fibonacci (loop)	88.12%	87.70%	56.61%	56.61%
Prime	89.40%	89.01%	27.27%	27.27%
Knight Problem	85.22%	84.74%	68.45%	68.45%
Fibonacci (rec.)	87.39%	87.11%	30.25%	30.25%
Hanoi towers	87.04%	87.01%	42.68%	42.68%
Average	87.42%	87.10%	41.76%	41.76%

Dictionary size: 16 entries

Comparison with other Pre-T tracers

- Our tracer achieves the longest trace cycle in Pre-T trace mode
  - 100 % circular utilization
  - Maintain the same compression ratio as forward encoding
  - 22% better trace cycle than the periodical triggering approach

	Existing industrial case (AHBTRACE)	Periodical triggering tracer	Our reverse encoding tracer
Effective trace buffer utilization	100%	93.8%	100%
Effective traced cycles	90	358 (3.98x)	437 (4.86x)

•Assume 8 segments for periodical triggering
•Circular buffer size: 1 KB

Comparison with other Pre-T tracers

- Our reverse encoding tracer has significant small hardware overhead
  - Only 6% hardware overhead
  - The periodical triggering tracer incurs 46% hardware overhead
    - Ping-pong organization
    - Complex circular buffer management

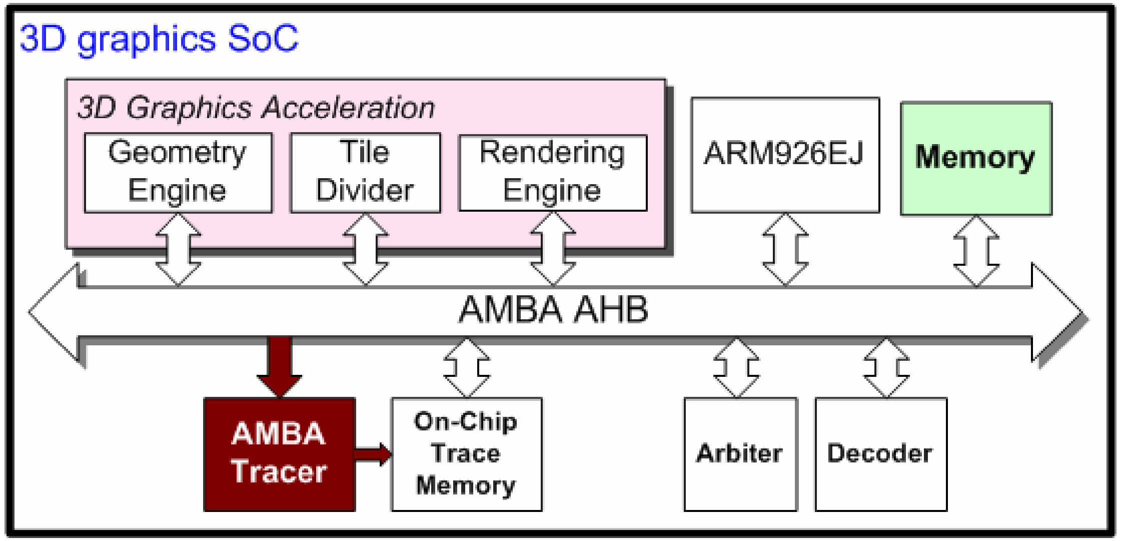
	Base tracer	Periodical triggering tracer	Our reverse encoding tracer
Area (Gate count)	41,768	60,783 (+46%)	44,447 (+6%)
Frequency (MHz)	500	500	500

•Base tracer: forward encoding based, only support Post-T trace
•Dictionary size: 16 entries
•Under TSMC 0.13 μm technology

Synthesis example in 3D graphics SoC platform

- The gate count is not huge in a typical SoC
  - 44 k gate count
- It is not the critical path

	3D Graphics SoC	Bus tracer
Area (Gate count)	~700,000 gates	44,447 gates
Max frequency (MHz)	233 MHz	500 MHz



Use 0.13 μ m technology with TSMC cell library

Summary

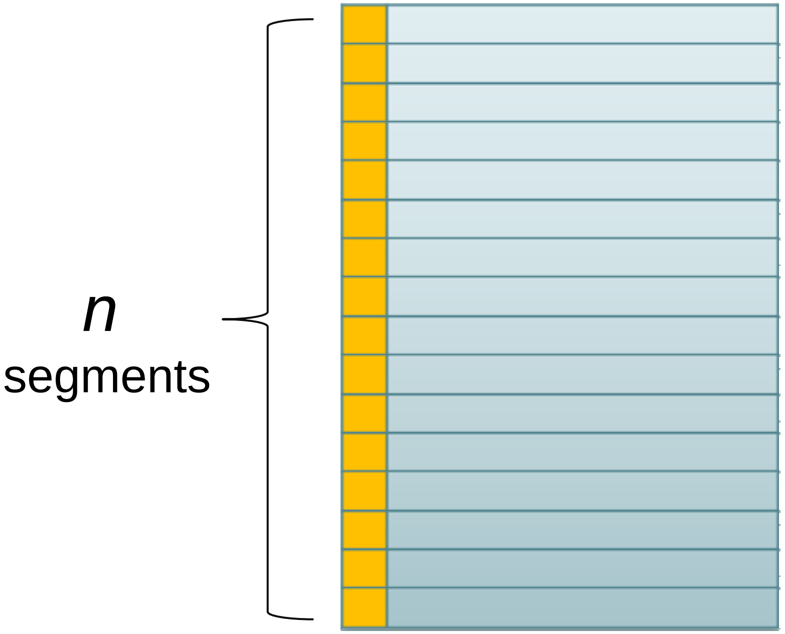
- Traditional forward encoding faces difficulties when wrapping around occurs in Pre-T trace
- A *reverse encoding algorithm* is proposed
  - It applies to data compression based on data relationship
  - Solve the problem caused by wrapping around
  - Maintain the same compression ratio as the forward encoding
- A realtime on-chip AHB bus tracer supporting both the Post-T/Pre-T trace
  - Significant small hardware overhead (6%)
  - Improve the previous approach significant
    - 22% better trace cycle
    - 40% less hardware overhead
- Future work
  - Extend it to other advance bus, *e.g.*, AXI and OCP.

Thank you!

Backup Slides

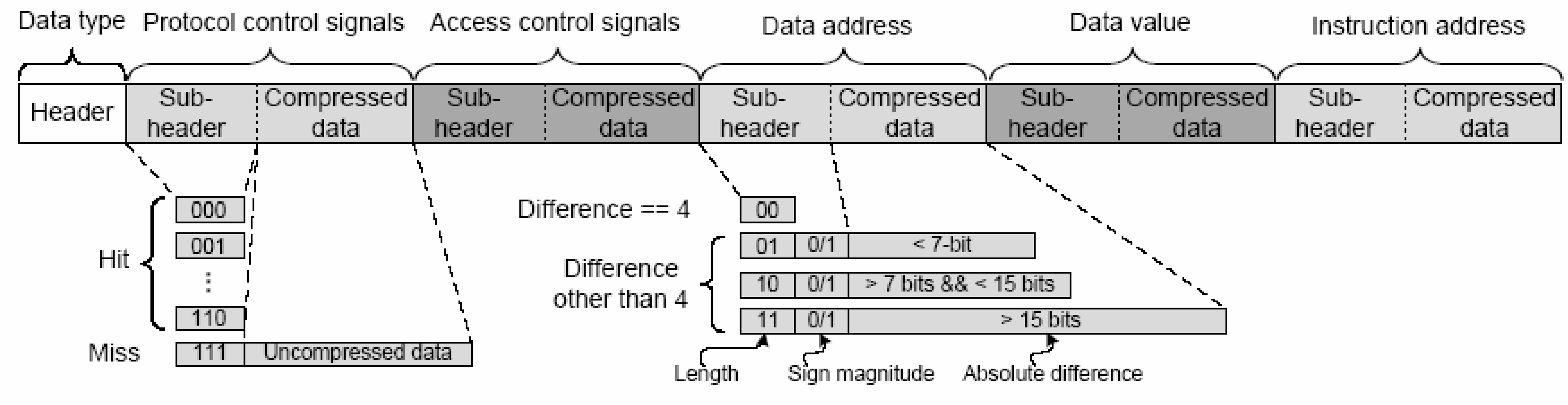
Drawbacks of periodical triggering

- Inefficient circular buffer utilization
  - Limited to $\frac{n-1}{n}$
- Decreased compression ratio
  - More initial uncompressed data
- Hugh hardware overhead



Circular buffer

Packet format



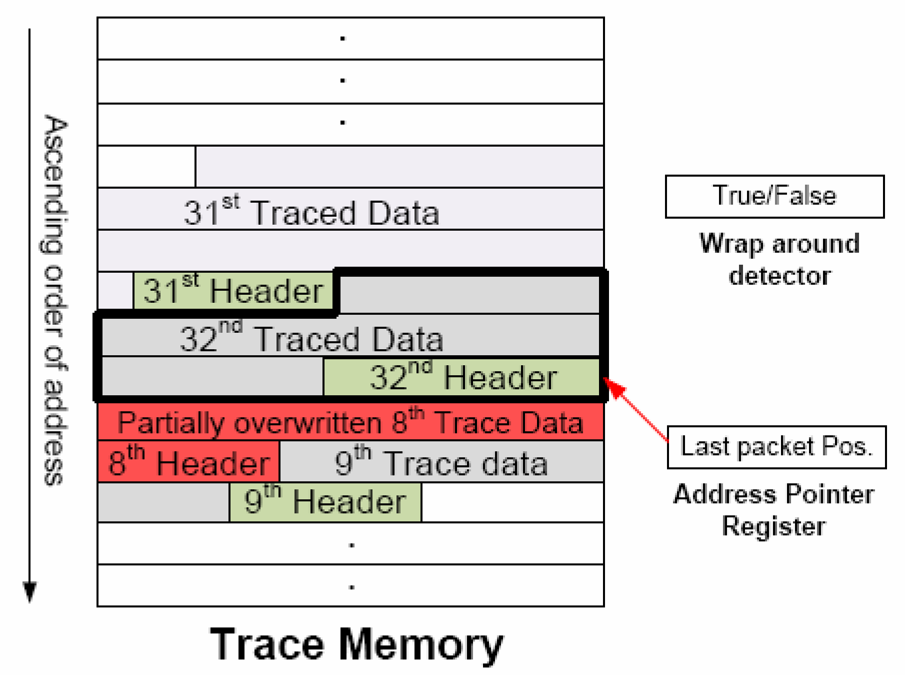
Data type
Protocol control signals
Access control signals
Data address
Data value
Instruction address

Header
Sub-header
Compressed data
Sub-header
Compressed data
Sub-header
Compressed data
Sub-header
Compressed data
Sub-header
Compressed data

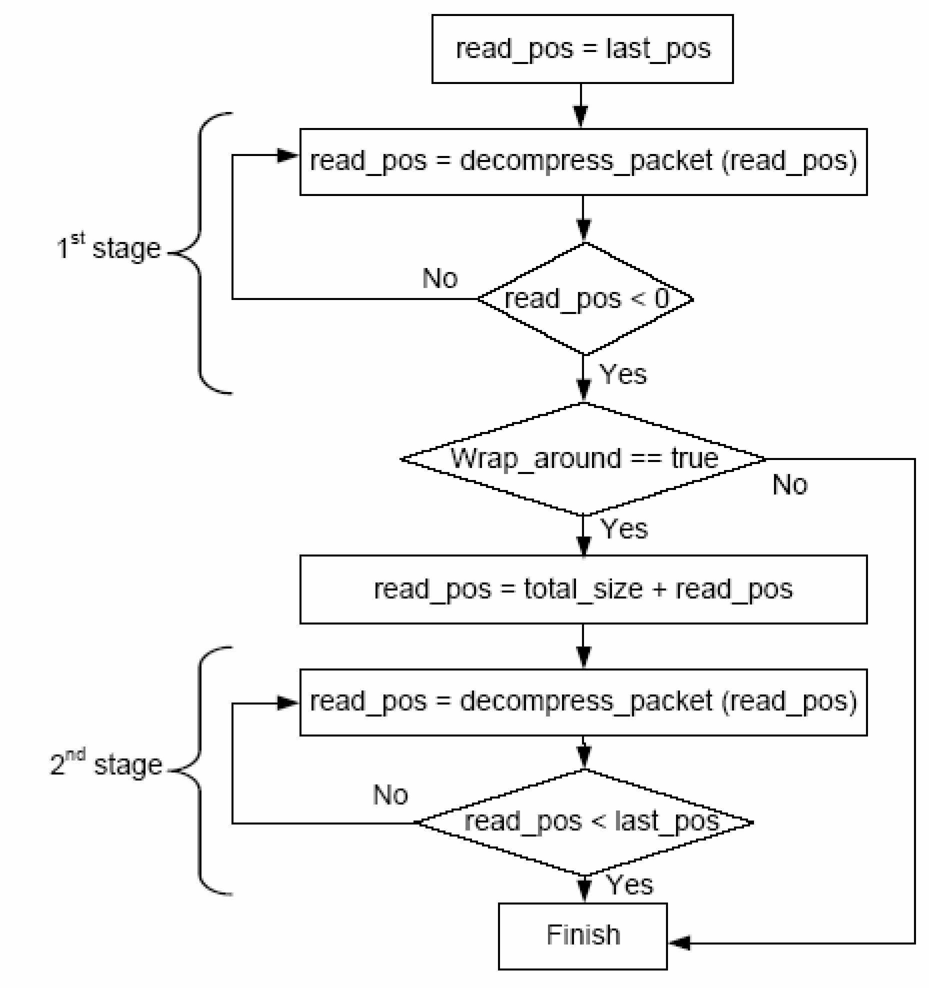
Hit
000
001
⋮
110
Miss
111
Uncompressed data

Difference == 4
00
Difference other than 4
01
0/1
< 7-bit
10
0/1
> 7 bits && < 15 bits
11
0/1
> 15 bits
Length
Sign magnitude
Absolute difference

Circular buffer management



Decompression flow


read_pos = last_pos
read_pos = decompress_packet (read_pos)
1st stage
No
read_pos < 0
Yes
Wrap_around == true
No
Yes
read_pos = total_size + read_pos
read_pos = decompress_packet (read_pos)
2nd stage
No
read_pos < last_pos
Yes
Finish