Motivation

• Today’s designs may have tens of thousands of memories with repair redundancy
• It takes a long time to load the repair data serially during system power-up
• Can we take advantage of the fact that very few of the memories actually need repair to significantly speed up the process?
• The general memory repair system
• Prior work
• Configurable BISR chain repair system
• Experimental results
• Conclusions
The general memory repair system

- Dedicated repair register for each memory
- Repair enable indicates that memory needs repair
- Most repair registers contain only contain 0s and allow compression of repair information in fuse box
Prior work (repair sharing)

- Use same repair solution for several memories
- Good results obtained for memories using row repair and memories behind a shared bus
- Limited application for distributed memories using column repair
- Potential yield loss
Prior work (memory bypass)

- Each repair register can be bypassed
- Configuration chain loaded first to select repair registers to include in chain
- Pipeline flop needed to avoid long asynchronous paths
- Speedup limited to about 5X
Configurable BISR chain repair system

• Extend Devanathan’s idea to bypass several repair registers at a time
• Bypassing longer segments reduces the overhead associated to the configuration chain
Segment selection circuit (SSC)

- Structure similar to Segment Insertion Bit (SIB) of IEEE 1687
  - Selects/bypasses associated chain segment
- SSC has additional circuitry to identify segments that need repair
  - 1-detection logic
Active scan path with bypassed segment

- Left segment included in scan path because at least one memory needs repair
- Right segment bypassed because none of the memories need repair
  - Segment input forced to 0
Active scan path (configuration chain selected)

- All segments are bypassed to load chain configuration
Repair data programming sequence

1. Apply Power-up sequence
2. Run Memory BIST
3. Transfer repair data from BIST controller to BISR register
4. Run 1-detection to generate the segment selection data
5. Rotate the configuration chain to program the segment selection data to the fuse box
6. Config the BISR chain based on the segment selection data
7. Calculate the new BISR chain length by injecting 1
8. Transfer repair data from BIST controller to BISR register
9. Program the repair data of selected BISR segment to the fuse box
Power-up sequence

1. Reset BISR chain and SSC
2. Inject leading 1 and load the segment selection data to the configuration chain
3. Update configuration chain length and apply segment selection data to config the BISR chain
4. Reset reg0 of SSC
5. Inject leading 1 and load the repair data to the BISR chain
6. Update the BISR chain length
Partitioning algorithm considerations

• Number of segments depends on a few factors
• Most important one is defect density
  – High defect density requires shorter segments to reduce probability of having to include a segment
• Segments of pre-existing IP blocks must be included as is
  – Not always possible to implement optimal segment size
Calculation of optimal segment size

• The BISR Chain Shifting time \( T = N_{repair} \times \frac{L}{N_{seg}} + 2 \times N_{seg} \)
  – \( L \): the total length of the repair registers
  – \( N_{seg} \): number of segments
  – \( N_{repair} \): number of segments requiring repair
• To minimize \( T \)
  – \( (N_{repair} \times L / N_{seg} + 2 \times N_{seg})' = 0 \)
• Optimal number of segments: \( N_{seg} = \sqrt{N_{repair} \times L / 2} \)
• Optimal Segment size \( N_{size} = L / N_{seg} \)
Repair data loading speedup factor (single repair)

Speedup factor vs. Generic

Speedup factor vs. Devanathan

# of memories
Repair data loading speedup factor (two repairs)

vs. Generic

vs. Devanathan

Speedup factor vs. # of memories

- M2/M3
- M1/M3
Repair data loading cycles (assumed vs actual number of repairs)

Reference: Generic method requires 100110 clock cycles
Conclusions

• A configurable BISR chain repair system is proposed to speed up repair data loading during chip power-up

• Experimental results show that number of clock cycles can be reduced by one to two orders of magnitude compared to previous methods