Adaptive Robustness Tuning for High Performance Domino Logic

Bharan Giridhar\textsuperscript{1}, David Fick\textsuperscript{1}, Matthew Fojtik\textsuperscript{1}, Sudhir Satpathy\textsuperscript{1}, David Bull\textsuperscript{2}, Dennis Sylvester\textsuperscript{1} and David Blaauw\textsuperscript{1}

\textsuperscript{1}University of Michigan, USA
\textsuperscript{2}ARM, United Kingdom
**Motivation**

- Targeted use of Domino logic remains the designer’s choice to implement speed critical paths in power constrained designs.

  *Samsung Hummingbird, 2009  
  http://www.samsung.com*

  *AMD Bulldozer  
  Golden et al., ISSCC 2011*

- With increasing process variation, domino logic is getting less beneficial:
  - Increasing margins for leakage, charge sharing and noise
Motivation

• Margining the keeper for robustness under worst-case PVT can degrade performance by ~32%

• ART Domino tracks PVT and trades extra robustness and timing margins for performance
ART Domino Gate

- Fast, speculative evaluation followed by fully margined (safe) evaluation to detect errors
ART Domino Gate

- Fast, speculative evaluation followed by fully margined (safe) evaluation to detect errors
ART Domino Gate

- Fast, speculative evaluation followed by fully margined (safe) evaluation to detect errors

Headers/footers shared across gates to minimize overhead
ART Domino Gate

- $V_X < VDD$ and $V_Y > VSS$ speed critical transitions at both nodes by reducing voltage swings.
- $V_Y > VSS$ speeds the following gate by trading its noise margin for speed.

Remove margins
ART Domino Gate

- $V_X < VDD$ and $V_Y > VSS$ speed critical transitions at both nodes by reducing voltage swings
- $V_Y > VSS$ speeds the following gate by trading its noise margin for speed

Fast, speculative evaluation
ART Domino Gate

- Slower, safe evaluation performed in the background
  - No impact on computation latency
ART Domino Gate

- Slower, safe evaluation performed in the background
  - No impact on computation latency
ART Domino Gate

- In case of errors, the errant computation is flushed from the pipeline
  - The result of safe evaluation is propagated, guaranteeing forward progress

![Error Detection Diagram]

---

`Slide 10`
ART Pipeline

Conventional Domino Pipeline

Pipe Stage N

Pipe Stage N+1

Overlapping clocks

Latchless hand-off
ART Pipeline
• During SE, DOMBUF snoops on the value propagated forward through the mux
• Overlapping clocks eliminate latches between pipe stages, provide skew tolerance
• During SE, DOMBUF snoops on the value propagated forward through the mux
• Overlapping clocks eliminate latches between pipe stages, provide skew tolerance
To allow the slower safe evaluation to complete, each pipe stage is split in half during CE.

Value stored on DOMBUF propagated forward cutting the stage depth by half.
Both halves of each pipe stage perform safe evaluations simultaneously.
• Both halves of each pipe stage perform safe evaluations simultaneously
• Error detector at each DOMBUF checks the segment till the preceding DOMBUF for errors
ART Pipeline

Error Detection

- CP: Precharge error logic (in red)
- CE: Copy Gate4 TO BUF2 and DOMBUF to BUF1
- SP+SE: Evaluate Domino XOR, Domino OR tree
ART Pipeline

DOMBUF, BUF1(2) are fully margined
ART Clock Generation

Domino gates clocked by the global clock generator

Power gates and other locally derived signals clocked using locally generated clocks
32 × 32b multiplier in 65nm CMOS
4 TVDD/TVSS voltage domains, 2 pipeline stages
Die Micrograph

Die Size: 1496μm X 1496μm

65nm CMOS process
Measured Results

- Measured Max. Performance: 1192 MHz

- Performance with ART up by 34% by eliminating robustness margins at nominal PVT

- Error region (Robustness failure)
Measured Results

• Method applicable to performance constrained logic where performance cannot be obtained by any other means

• Power increases sharply at higher frequencies due to increased short circuit current on the output inverter

Minimum Power at each frequency

ΔTVDD/ΔTVSS at each measured power-frequency point

Simulated Power
Standard Domino

---

No Adapting

~34% performance gain with
12% Power increase

~17% Performance gain with
12% Power reduction

---

850 900 950 1000 1050 1100 1150 1200

Performance (MHz)

0.00
0.05
0.10
0.15
0.20
0.25
0.30

Required ΔTVDD (V)

850 900 950 1000 1050 1100 1150 1200

Required ΔTVSS (V)

850 900 950 1000 1050 1100 1150 1200

Simulated Power
Standard Domino

---

No Adapting

~34% performance gain with
47% Power increase

~17% Performance gain with
12% Power reduction

---

Performance (MHz)

Power (mW)
The measured average % gain due to robustness tuning across 20 dies is ~28%.

% Gain decreases at higher temperatures as gates become less robust.

At 27°C, 1.2V VDD:

- 20 Dies

% Performance Gain (Robustness Speculation)
**Overall Performance Gain**

Total gains of 49% to 71% over conventionally margined designs.

* Worst process was set by the slowest die. Temperature was set to 85°C and Supply was degraded by 10% to 1.08V.
Error rate is more sensitive to TVSS than TVDD
  – TVSS tuning also affects robustness of following gate

\[ \Delta_{TVDD}(V), \Delta_{TVSS}(V) \]

\[ \Delta_{TVDD} = 0.9V-TVDD, \quad \Delta_{TVSS} = TVSS-0.17V \]
Conclusion

• A new high speed design style called ART Domino was presented
• ART Domino provides overall gain of up to 1.71x over conventional domino
  – 3.2x faster than static CMOS
• Design overhead is amortized over pipeline stage depth
Thank You

Questions?
ART Clock Generation

Globally generated clocks with relaxed skew constraints

Locally generated clocking signals with stricter skew constraints

Example
4-Stage Pipeline Operation Phases
System Implementation
Conventional Domino

Other Logic → FF → Conventional Domino Logic → FF → Other Logic
System Implementation
Conventional Domino

STG1

Eval
Prchg
System Implementation
Conventional Domino

STG2

Prchg  Eval

Slide 35
System Implementation
ART Domino

Other Logic -> FF -> ART Domino Logic -> FF -> Other Logic
System Implementation
ART Domino
System Implementation
ART Domino

[Diagram showing System Implementation with labels CP, SE, SP, CE and STG2]
System Implementation
ART Domino

STG3

CE  CP  SE  SP
System Implementation
ART Domino

STG4

SP CE CP SE
System Implementation
ART Domino

SYSCLK (GCLK/2)

STG_1

SE CP CE SP SE

STG_2

SP SE CP CE SP SE

STG_3 (=STG_1 CLK)

CE SP SE CP CE SP SE

STG_4 (=STG_2 CLK)

CP CE SP SE CP CE SP SE

Error Detect STG1

Error Detect STG1

Error Detect STG1

Error Detect STG1
Architectural Replay

Cycle0

STG_1

SE CP CE SP

STG_2

SP SE CP CE

STG_3

CE SP SE CP

STG_4

CP CE SP SE

Cycle1

Error detected in STG2

Cycle2

SE CP CE SP

STG_1

SP SE CP CE

STG_2

CE SP SE CP

STG_3

CP CE SP SE

STG_4

CE SP SE CP

SYSCLK (GCLK/2)
Error detected in STG2

Copy safe value
Architectural Replay

Cycle0

Cycle1

Cycle2

Error detected in STG2

Copy safe value

SYSCLK (GCLK/2)

STG_1

STG_2

STG_3

STG_4

SE  CP  CE  SP  SE

SP  SE  CP  CE  SP  SE

CE  SP  SE  CP  CE  SP  SE

CP  CE  SP  SE  CP  CE  SP  SE
Error Detection Scenarios

- Incorrect: Standard error detection
- Correct: False errors flagged, but functionality maintained

Special case; Requires additional check

Extra comparison between LAT_IN and LAT_OUT