Simple switched GALS interconnect

George Taylor
Robert Mullins
Simon Moore

Contents

• System overview
• Project requirements
• Memory system interface
• Simple interconnect
• GALS wrapper
• Memory system internals
• Token ring
Physical picture

microcontrollers

SRAM + memory controller

i/o devices

System overview

- DI interconnect, delay matched SRAM + IO + off-chip
Some motivation

- Projects
  - Smart card security
  - Self-timed microcontroller study
- Ring arbiter
  - Lazy arbiter good for common case of just one processor
  - Alternative would be simple multiplexer
  - One processor can issue random memory requests (security)
- Simple pipelining in memory system
  - Original microcontroller has separate program and data memories, combined into one on this chip
  - Reduces cost of random memory requests
- 1-of-4 interconnect
  - Delay insensitive without concern for performance permitted very fast top level assembly
  - Hides data dependent power, fewer transitions than 1-of-2

Memory controller interface

- Bundled data circuit in ‘middle’ of DI block
  - Matching delay is selected depending on operation and is configurable
- Commands, addresses and data enter on left, read results exit on right
- Pipeline remembers which processor issued the request so read results can later be routed correctly (not needed for write)
  - pnum is 1-of-n
Simple interconnect – I

• Inverted 1-of-4 used
  – \((1111, 1110, 1101, 1011, 0111)\)
• Merge

\[
\text{addr0}[i] \ldots \text{addr4}[i] \rightarrow \text{addr}[i] \rightarrow \text{ackin0}
\]

\(x32\) for 16 bit address
similar for datain0..4, iswrite0..4, but not isread0..4 (pnum)

\[
\text{C} \rightarrow \text{isread0} \rightarrow \text{iswrite0} \rightarrow \text{ackin0}
\]

\(x5\) for 5 processors

• remember arbitration has already been performed

Simple interconnect – II

• Demux/switch

\[
\text{pnum}[p] \rightarrow \text{dataoutP}[i] \rightarrow \text{dataout}[i]
\]

\(x36\) for 18 bit data, \(x4\) for processor P 0...4

– straightforward
– this is a large number of C elements
  • some assumptions were made and the C elements replaced with simpler gates
– acknowledgements from processors to memory merged using OR gate
GALS wrapper – I

- Simple ring oscillator with pausable clock
  - stop input to disable clock or pause whilst adjustment is performed (counter not shown)
  - stopped signal indicates when clock stopped and hence delay can safely be adjusted, verified

GALS wrapper – II

- Synchronous microcontroller clock paused for memory access
  - no pipelining
Memory system

Token ring – I

- three wires in ring
- low component count
- fast request/grant when token already held
  - mutex + AND gate delay
Token ring – II

- Simple extra logic added for reset and initialisation, each arbiter can be
  - held in reset without token
  - held in reset with token
  - release reset without changing token status
- Formal verification against simple specification
  - using in-house Veraci tool
- Problem
  - timing problem (poor assumption) in memory system prohibits use of multiple processors
  - only “one-shot” testing of ring arbiter possible

Summary

- Example design connecting several microcontrollers to a common memory with:
  - 1-of-4 wiring
  - simple merge and switch (demux)
  - GALS wrapper
  - ring arbiter
- Deployed on Springbank test chip