Modeling and Analysis of the System Bus on the SoC Platform

Eun Ju Choi, Young Sin Cho, Kyoung Rok Cho

Communication Circuits and System Design Lab.
Chungbuk National University
12, Gaeshin-dong, Cheongju-city, Rep. of Korea
Contents

- SoC platform & shared-bus
- Design issue
- Proposed latency model
- Simulation & result
- Conclusions
Contents

- SoC platform & shared-bus
- Design issue
- Proposed latency model
- Simulation & result
- Conclusions
Platform based SoC design

- Design methodology, Verification environment, ...etc
- IP reuse
- To reduce cost, time, effort.

Pre-verified IP DB

Hardware IP

Software IP

Programmable

Costumer Specific IP

- Any CPU core
- CPU Bridge
- SRAM
- On-chip Memory Controller

Interconnection

- Power Manager
- RTC
- Interrupt Ctrl
- Watch Dog
- Program I/O
- I^2C
- UART
- DMA
- GT Timers

< Hardware Platform >

Shared bus for interconnection

- Simple architecture
- Totally reusable
- Lower speed than resident cores
- Performance depends on an arbitration
- Efficient solution in the current design flows
Single-layer bus

- Number of IPs on a bus
- Only one master grabs a ownership at a time
Multi-layer bus

- Multi-path between master and slave
  - Each layer can be simple
  - Increase bandwidth

- Weak point
  - Hardware resource
  - Power
  - Design complexity
AMBA: popular standard for SoC

- Open standard, on-chip bus specification by ARM
- AHB, ASB, APB, AXI
- Support multi-layer architecture
- Advanced High-performance Bus
  - Pipelined operation
  - Non-tristate implementation
  - Multiple bus masters
  - Burst transfers
  - Split transactions

Diagram:

- High-performance ARM Processor
- High-bandwidth on-chip RAM
- High-bandwidth External memory interface
- DMA bus master
- UART
- Timer
- Keypad
- PIO
- AHB or ASB
- APB
Design issue

How can you estimate a throughput from the present shared-bus before actual design?

- Number of masters
- Number of layers
- Transfer properties
Contents

- SoC platform & shared-bus
- Design issue
- Proposed latency model
- Simulation & result
- Conclusions
IS(Ideal-Slave) latency model

- A slave has no latency to response to a master.
- $L_{Bus}$ – Latency of shared-bus
- $L_{Complex\_Bus}$ – Latency of shared-bus including multiple master
- $L_{Single\_Layer}$ – Latency of single-layer bus
- $L_{Multi\_Layer}$ – Latency of multi-layer bus

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_M$</td>
<td>Number of masters</td>
</tr>
<tr>
<td>$N_L$</td>
<td>Number of layers</td>
</tr>
<tr>
<td>$N_D$</td>
<td>Number of data</td>
</tr>
<tr>
<td>$S$</td>
<td>Single transfer ratio</td>
</tr>
<tr>
<td>$B$</td>
<td>Burst size</td>
</tr>
<tr>
<td>$U$</td>
<td>Usage of bus</td>
</tr>
<tr>
<td>$A$</td>
<td>Active bridge ratio</td>
</tr>
</tbody>
</table>
Latency for shared bus

\[ L_{\text{Bus}} = 1 + N_D \]  

(1)

where \( N_D \) is number of data and ‘1’ indicate the request cycle getting approval from a bus arbiter.

Consider an effect of transfer mode and pipelined architecture

\[ L_{\text{Bus}} = 3 \cdot N_D \cdot S \\
+ \left\{ \text{Ceiling}\left( \frac{N_D \cdot (1 - S)}{B} \right) + N_D \cdot (1 - S) \right\} \]  

(2)

where \( S(0 \leq S \leq 1) \) is a ratio of single transfer and \( B \) is a burst data size.

<An example of two transfer type in shared bus>
Latency for single-layer shared bus

\[ L_{\text{Single\_Layer}} = N_M \cdot L_{\text{Bus}} \]  \hspace{1cm} (3)

\( N_M \) is the number of masters.

- All master IPs are connected to the single layer bus and are controlled by an arbiter.
- This one master IP latency occupies the shared bus.
Latency for single-layer shared bus

\[ L_{\text{Single\_Layer}} = N_M \cdot L_{\text{Bus}} \]

\[ L_{\text{Bus\_Complex}} = \left( 3 - 2 \cdot U \right) \cdot N_D \cdot S \]
\[ + \left\{ \text{Ceiling} \left( \frac{N_D \cdot (1 - S)}{B} \right) + N_D \cdot (1 - S) \right\} \]

where \( U(0 \leq U \leq 1) \) is usage of bus which is a probability of continuing single transfer.

- If two or more master I/Os are connected to the bus, address and data cycle access the bus simultaneously.
- The effect of the pipeline architecture depends on the bus usage.

<An example that shows two master transfer the data continuously>
A partition of bandwidth according to number of masters

- Increase in bus usage means increase the probability of the continuing data processing.
- Total bandwidth is equal to total bandwidth of each master IP.
Latency for multi-layer shared bus

\[ L_{\text{Multi\_Layer}} = \frac{N_M}{N_L} \cdot L_{\text{Bus\_Complex}} \cdot (1 - A) + \alpha \cdot A \] (5)

where \( A(0 \leq U \leq 1) \) is a probability making a data path through a bridge module. Bridge factor, \( \alpha \), is latency overhead caused by bridge module. \( N_L \) is number of layers.

The multi-layer structure with bridge module:
Latency for multi-layer shared bus

- The latency is increased due to bridge modules.
- If two layers are connected through a bridge module, one IP should be a master of both layers.
- It cannot offer entire bandwidth of two layers.

(a) Case of 0 bridge activated
(b) Case of 1 bridge activated
(c) Case of 2 bridge activated

<The configuration of data path with 3-layer bus>
Latency for multi-layer shared bus

Data paths which use same number of bridge modules

\[
\alpha = \sum_{i=1}^{N_l-1} \binom{N_l-1}{i} \cdot \frac{N_M}{N_l-i} \cdot L_{Bus\_Complex}
\]

(6)

Total number of data paths using bridges which can may appeared on multi-layer bus

- Bridge factor is the latency overhead by the using bridge.
- Bridge factor depends on the number of data path.

<The distribution of probability A by combination of data path>
Throughtput ratio of multi-layers to single-layer

- the throughput is inversely proportional to A and proportional to number of layer
Contents

- SoC platform & shared-bus
- Design issue
- Proposed latency model
- Simulation & result
- Conclusions
Result of latency model for shared bus ($N_D = 1000$)

- Case 1) If same $U$ (bus usage), $S \uparrow$, latency $\uparrow$.
- Case 2) If same $S$ (single transfer rate), $U \uparrow$, latency $\downarrow$.

If the system which has high $U$, it doesn’t have to much consider about $S$. 

<The variation of latency according to increase of bus usage and single transfer>
Simulation & Result (2/9)

Result of latency model for shared bus

- The latency is reduced when compare multi-layers with single-layer. (2-layers 45% ↓, 3-layers 63% ↓)

- The condition (S, B, U, A) depends on characteristic of SoC.

<The latency difference of each shared bus by parameter number of master IPs>
Result of latency model for multi-layer bus

Condition: VGA(640x480), 30frame/s
Job requirement: 27.65[MB/s] = 640X480X3[Byte]X30[frame]
Total requirement: 110.6[MB/s] = 27.65[MB/s]X4

Max throughput of USB2.0 is 480[Mb/s] (= 60[MB/s])

<The simple example of image processing by MPEG>

<The expected throughput of each shared bus according to increase number of masters>
We use MaxSim for a comparison of simulation results

- Modeling & simulation tools for SoC designs
- Cycle-accurate models

(a) single-layer architecture

(b) multi-layer architecture

<The example of SoC on the MaxSim with single-layer and multi-layer>
Single-layer results

- \( N_D = 1000 \)
- 96% accuracy

<The comparison of the results between IS model and MaxSim>
Simulation & Result (6/9)

2-layer results
- \( N_D = 1000, A = 20\% \)
- 85% accuracy

The comparison of the results between IS model and MaxSim
Simulation & Result (7/9)

3-layer results
- $N_D = 1000$, $A = 20\%$
- 85% accuracy

*The comparison of the results between IS model and MaxSim*
The accuracy of the proposed latency model are over **96%** for single-layer and **85%** for multiple layers.
- The bus usage indicates an average utilization of the bus as function of number of master IPs
Contents

- SoC platform & shared-bus
- Design issue
- Proposed latency model
- Simulation & result
- Conclusions
Conclusions

We propose a latency model (IS model) which to estimate a performance of system bus before actual design.

Simulation & result

- Analyze the parameters of shared bus latency
- Analyze number of masters affecting to bus throughput
- Find out an appropriate number of layers on specific SoCs
- Compare the results with that of MaxSim
Thank you!