ECE 720
Electronic System Level & Physical Design

Dr. W. Rhett Davis

Project 2
Anusha Ravilla

Introduction

Transaction Level Modelling is an extension of SystemC, TLM simplifies modeling of systems which is quite complicated with SystemC. SystemC and TLM are high level modelling strategies. At the TLM level, the functionality of transfers is modelled. TLM allows the designer to tinker with the communication between modules without making changes to the RTL. The generic payload class in the TLM helps to contain the information needed for a transaction through a memory-mapped bus. Since the generic payload gives a generalized way of defining, it helps in reusing of code independent of the bus. The simulation speed. This project aims to show the speedup in cycles/sec performance of TLM simulations compared to RTL simulations of the Cortex-M0DS SoC running the program dma.c. Additionally, it is expected that a comparison between the simulation time of TLM to that of RTL results in a negligible prediction error.

Implementation Method

The first step in the implementation is introduction of the dma unit (dma_module) between the ahb_1 and the memory controller. The architecture is such that the transactions sent via the ahb bus are sent into the DMA module and then sent to the memory controller devoid of whether the transaction aims to use the DMA. This is achieved by making appropriate changes to the initiatorBTransport function in SimpleBusLT.h.

The second step in the implementation is modelling the latency. The latency of reads and writes differ based on whether they use the DMA or they just bypass the DMA. The latency is also affected by the sequence in which the rows and banks are accessed.

The custom_b_transport gets the generic payload as the input and this function was modified to model the delays. The challenge faced while modelling the delays was the figuring out if the instruction was DMA bound local to this function. The byte_enable_length helped resolve this problem.

The delays were modelled as follows:

Delay for initial access of a bank (read): 80 cycles.

Delay for successive access' to the same row and bank(read): 60 cycles

Delay for successive access' to different row (read): 110 cycles

Delay for a write : 10 cycles

Delay for a DMA access : 100 cycles.

The simulation results obtained from the simulation of TLM is recorded in the following table:

TLM					
		AHB2 Xfer Count		Wall-clock	Performance
Sim Time (ns)	Cycles	Reads	Writes	time (s)	(cyc/s)
316390	31639	3836	290	0.41	77168.29268

The simulation results obtained from the simulation of RTL is recorded in the following table:

RTL					
		AHB2 Xfer Count		Wall-clock	Performance
Sim Time (ns)	Cycles	Reads	Writes	time (s)	(cyc/s)
178485	17848.5	2111	364	2	8924.25

It can be seen that the cycles per second performance of the TLM is much higher than that of the RTL. It is found to be 8.647 times that of the RTL simulation.

Predictions to Improve the TLM performance

Although the TLM simulation is undeniably faster than RTL since it completes in a much shorter time compared to the RTL simulation, all the other parameters observed in the above table indicate that the TLM can do better.

The number of reads in case of the TLM are 81.7 percent more than that of the RTL. The number of cycles and simulation time are significantly more than that of the RTL. Reads are expensive in terms of cycles, minimising the number of reads will help in improvement in the performance. TLM system calls might be a factor for the increased number of reads.

Precision Calculation

An increase of 77.3 percent from the RTL simulation time is observed for the TLM simulation. This will result in a very significant difference between the simulations which can raise suspicions on the accuracy of the TLM simulation. Therefore, a portion/ sequence if the simulation output is compared to find the precision.

However, in order to figure out the cause for the overall discrepancy, the code for RTL and TLM was compared, and it is observed that the code for TLM included for empty for loops which account for a significant number of cycles. An attempt to eliminate these loops resulted in an error in the simulation. The for loops are integral for the operation of the TLM but they cause delays.
Comparison of simulation time of specific sequences should prove to be precise as these loops occur as a chunk and sequences identified are not affected by them.

The following 4 sequences were identified and the simulation time of TLM and RTL was compared.

Sequence 1

The first occurrence of a transaction targeting the DMA is chosen , as this portion is unique and can be identified in both the simulations. This portion had 15 reads and 15 writes.
The number of cycles seen in the cassese of TLM and RTL and the error calculated between them can be seen in the table below:

TLM			RTL			
Sim Time (ns)			Sim Time (ns)			
Start	Stop	Cycles	Start	Stop	Cycles	Error
12750	14300	155	14195	15755	156	-1%

The precision observed is good but the sequence consists of a small subset.

Sequence 2

The aim to identify a larger sequence. A sequence of 41 reads is observed before the occurrence of the first dma instruction in the RTL as well as the TLM. The number of cycles TLM and RTL and the error calculated for this sequence is recorded in the following table:

TLM			RTL			
Sim Time (ns)			Sim Time (ns)			
Start	Stop	Cycles	Start	Stop	Cycles	Error
9430	12290	286	11025	13885	286	0%

The precision observed for this case of reads is very good.

Sequence 3

The sequences consider so far have been homogeneous and have been relatively short in length. So the next sequence considered should be longer. The next sequence is selected to span over the start of the simulation to the beginning of the Sequence 2. The number of reads are 133 and writes are 17. The number of cycles TLM and RTL and the error calculated for this sequence is recorded in the following table:

TLM			RTL			
Sim Time (ns)			Sim Time (ns)			
Start	Stop	Cycles	Start	Stop	Cycles	Error
0	9430	943	1995	11025	903	4%

The precision observed in this case is still acceptable. A combination of reads and writes in a sequence and increasing the length of a sequence leads to decrease in precision.

Sequence 4

The next sequence is just for a confirmation of what was observed out the earlier sequence 1. A second occurrence of DMA access is observed and the values are tabulated:

TLM			RTL			
Sim Time (ns)			Sim Time (ns)			
Start	Stop	Cycles	Start	Stop	Cycles	Error
78200	78980	78	16395	17175	78	0%

The precision observed in this case confirms with our inference that the precision is good for shorter simulations.

Conclusion

TLM simulation is observed to faster than the RTL simulation significantly which makes a case for the use of TLM. It is also more convenient to make changes in communication between models using TLM as compared to changing the RTL. However, the performance of the TLM can be improved further in terms of the number of reads and simulation time. The precision observed over the four sequences specified leads to an inference that the TLM provides acceptable accuracy.