Computer Architecture presentation

Name-Rahul Dhayal

Roll.No-B19EE069

topic-Hardware accelerators

A novel way to efficiently simulate complex full systems incorporating hardware accelerators

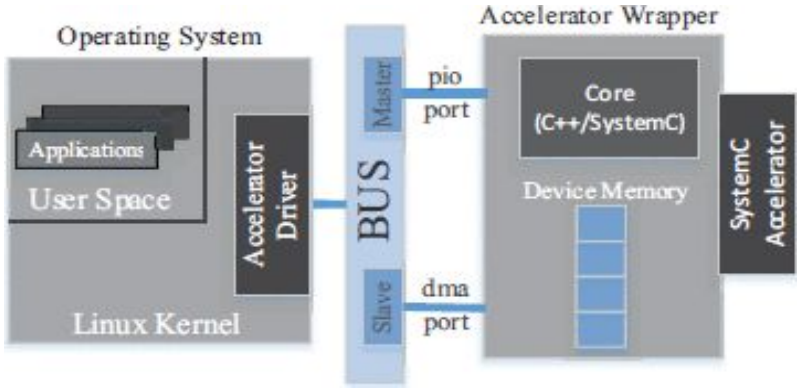
The idea

1. most simulators used for evaluating the complete user applications (i.e. full-system CPU/Mem/Peripheral simulators) lack any type of SystemC accelerator support.
2. We developed a novel simulation environment comprised of a full system simulator with a SystemC cycle-accurate hardware accelerator device.
3. Our System utilizes:

    a) GEM5 full-system simulator : simulates complete systems comprised of numerous CPUs,peripherals running full OS.

    b)Accellera(Open SystemC Initiative):provides an open-source proof-of-concept simulator.

Approach



1. **Operating System(OS)**

   a)a set of device drivers in Kernel mode have been developed

   b)our novel approach allows full overlap of the two sub-simulations

2. **Memory Bus**

   a)Bus master port is connected to the peripheral

   I/O Wrapper (write/read registers)

   b)Bus slave port is connected to the GEM5 DMA

   Wrapper (write/read large amounts of data)

3. Accelerator Wrapper

- A mixed C++/SystemC core -> connection of the GEM5 C++ functions and the accelerator's SystemC threads
- A large Device Memory to store the data from OS' memcpy was implemented (simulating the DDR memory of most of the real systems incorporating PCI- connected FPGA and/or GPU boards)

4. SystemC Accelerator

- a reference SystemC accelerator has been developed to evaluate the Accelerator Wrapper and the Linux Kernel Drivers
- Our approach is a helpful reference for any synthesizable SystemC accelerator

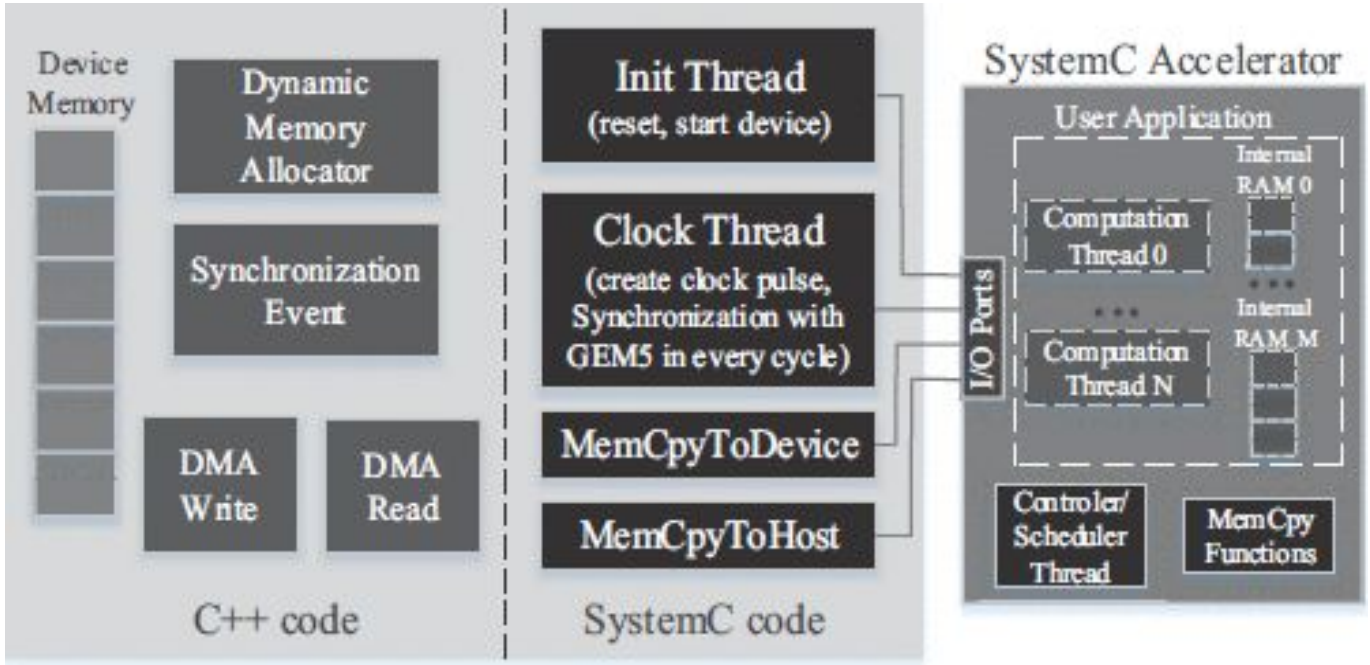


Fig. Accelerator wrapper device

Implementation

1. **Accelerator Wrapper**

   a)Buddy dynamic memory allocation algorithm scheme is implemented in the Accelerator Wrapper.

   b)2 DMA engines -> efficiently transfer high data volumes from the Linux driver to the Accelerator Wrapper and vise versa (cudaMemcpy similar)

   c)GEM5 synchronisation event ->triggered at every SystemC accelerator device cycle.It checks whether the SystemC accelerator has reached the next cycle

   d)Initialisation thread->generate the reset and start signals

   e)Clock thread-> generate the actual clock signal of the accelerator device cycle accelerator

   f)2 MemCpy SystemC threads-> pass the data from the Wrapper Device Memory to synthesisable Input/Output ports

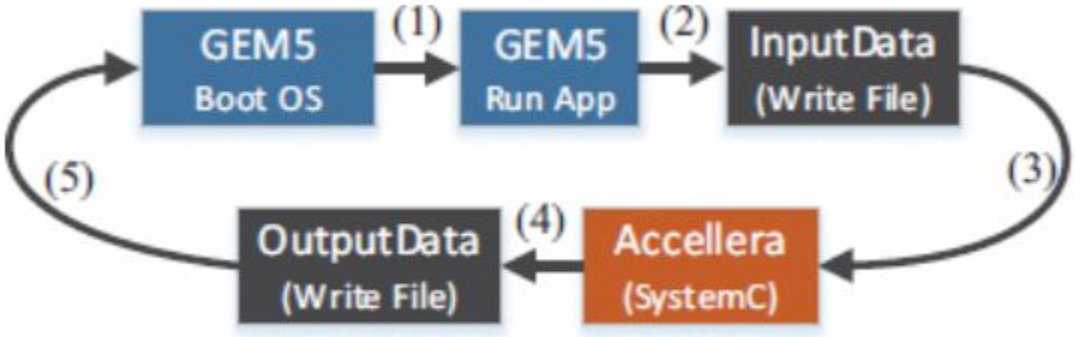
2. SystemC Accelerator

a)Controller/Scheduler SystemC Thread->This is the main SystemC thread which is called by GEM5's OS while the user has the ability to create as many individual cores as required

b)2 SystemC memcpy functions-> efficient communication with the Accelerator Wrapper Memory

Evaluation

- **Conventional Method**



a)In every SystemC simulator call, the full-system must boot the OS from scratch

b)no notion of synchronisation exists

- **Proposed method**

a)full-system simulator boots the OS only once

b)full global synchronization is supported through programmed Input/Output and DMA engines

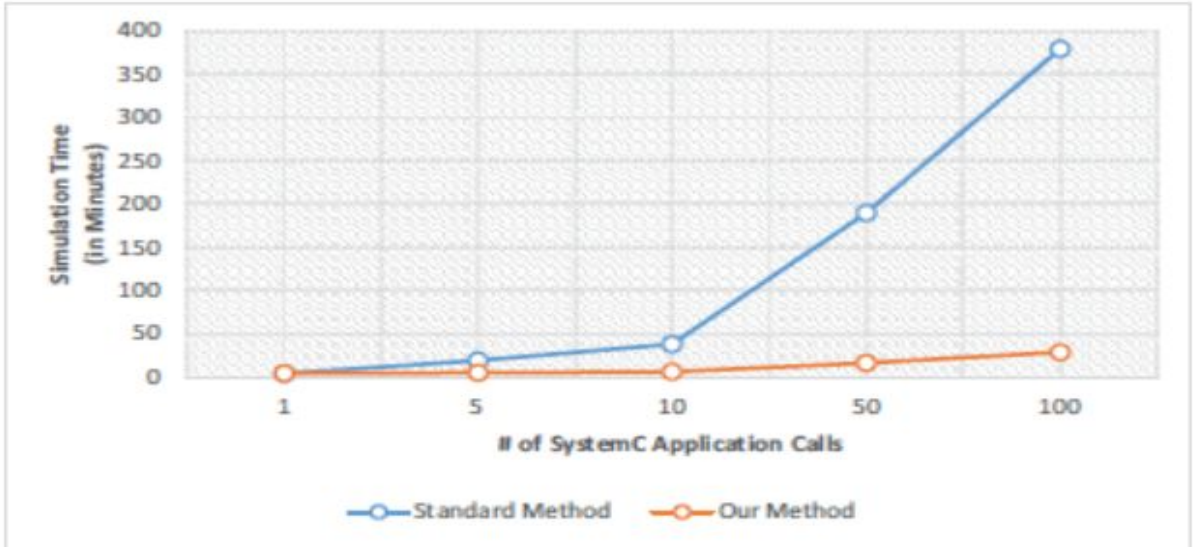
Results

System is evaluated on two use cases using Intel i7-3632QM at 2.2GHz with 8GB of RAM as processing platform.

Below table illustrates the overall simulation time when calling 10 times the SystemC accelerator by a standard method and our method.And in it we clearly see how much we accelerate our system.

Mutual Information			Ttransfer Entropy		
# of Iterations (Complexity)	Standard Method	Our Method	# of Iterations (Complexity)	Standard Method	Our Method
100	37.3 min	4.5 min	20	37.5 min	4.7 min
200	37.5 min	4.7 min	50	37.8 min	5.1 min
500	37.9 min	5.3 min	100	38.4 min	6 min

Below graph compares the overall simulation times between the conventional and the proposed method



Simulation time of TE using different # of accelerator calls

Conclusion

This paper proposes a novel approach for extending a full-system simulator with the ability to efficiently and accurately simulate hardware accelerator(s) designed in SystemC. Such an integrated approach can severally reduce the time needed for design space exploration while also accelerate the overall verification process.

THANK YOU