Software Pipelining Shader Core with Streaming Interface

Kwontaek Kwon[1*] Kyoung June Min[1] Derek Lentz[2] Sang Oak Woo[1] Seok Yoon Jung[1] Shi Hwa Lee[1]

[1]Samsung Advanced Institute of Technology [2]Samsung Information Systems America

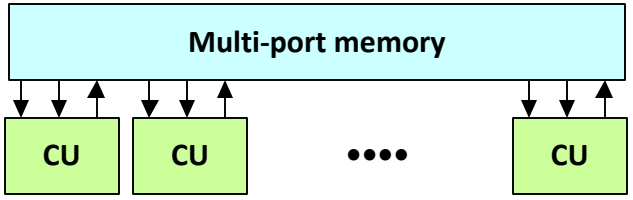
**kwontaek.kwon@samsung.com*

Motivation

Key factors to GPU computing performance

- Parallel operation of multiple computation units
- Parallel I/O bandwidth into/out of computation units

Large high-speed internal memory can be used to provide sufficient I/O bandwidth to computation units. But it incurs **significant hardware cost.**

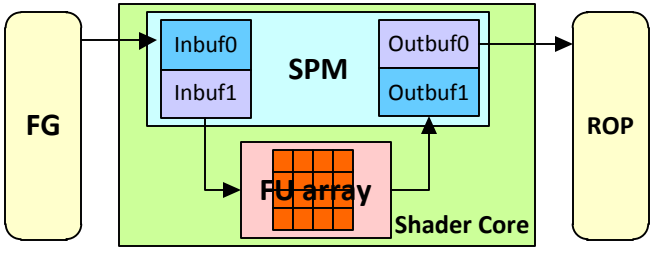


Shader Core Architecture based on SRAM internal memory

SRAM Internal memories in the shader core is simultaneously used for:

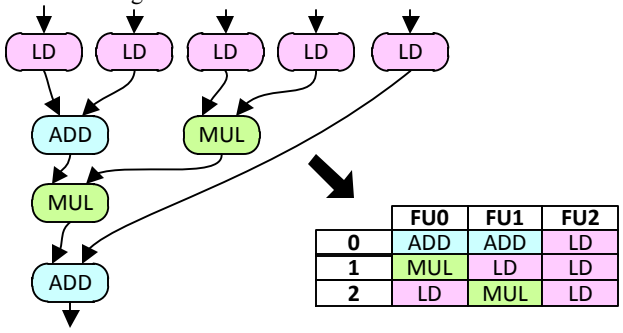
- Receiving pixel shader input data from external sources (e.g. Fragment Generator units)
- Providing shader input data to SRP FUs
- Storing shader output data batches
- Providing shader output data to external consumers (e.g. ROP units)

The SRAM-based architecture requires large, high-speed, multi-port, memories which are rather expensive in hardware and are often not able to provide sufficient I/O bandwidth to all FUs.



Software Pipelineing Shader Core

The Samsung Reconfigurable Processor(SRP)[1] can accelerate loop execution by software pipelining which overlaps computations from multiple iterations to make a compact schedule. However software pipelining would schedule multiple LD requests for an iteration with different timing and order.

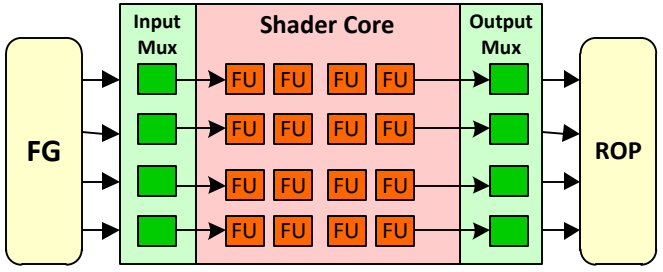


	FU0	FU1	FU2
0	ADD	ADD	LD
1	MUL	LD	LD
2	LD	MUL	LD

SRP GPU Architecture based on Streaming Interface

Streaming interface provides dedicated I/O channel between FU and external data source/destination:

- Each FU can do LD/ST with external source/destination simultaneously and continuously.
- Data routing and queuing are performed on-the-fly by the stream input/output muxes.
- Stream data LD/ST operations are automatically synchronized with software pipelining schedule.

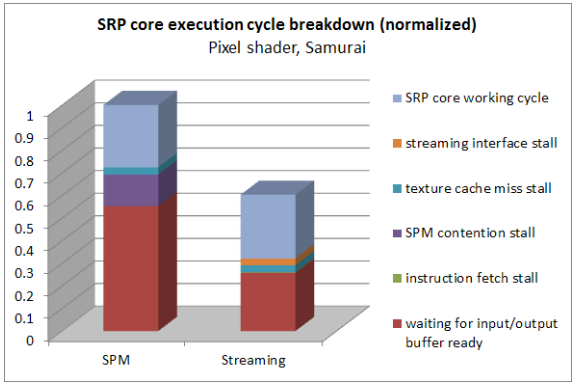


Evaluation and Conclusions

The SRP GPU architecture based on the streaming interface has following advantages:

- Eliminates FU I/O bandwidth limit imposed by SPM; each FU can have dedicated I/O channel.
- Required Internal memory size is much lower; Shader input/output batch buffer is not required since I/O is done in streaming manner.
- LD/ST operations with different timing and order are dealt by on-the-fly data routing and queuing.

We evaluated benefits of streaming interface-based architecture by measuring rendering time of Rightware's Samurai benchmark on FPGA verification environment.
The result shows upto 45% rendering performance gain and we can see that the performance bottleneck in SRP FU I/O is successfully removed.



References

[1] LEE W. J., WOO S. O., KWON K., SON S. J., MIN K. J., JANG G. J., LEE C. H., JUNG S. Y., PARK C. M., LEE S. H.: A scalable gpu architecture based on dynamically reconfigurable embedded processor. In *Poster. HPG '11* (2011).