Adaptive Computing in DAQ and Trigger Systems

Justus-Liebig-University Giessen

Ming Liu

Outline

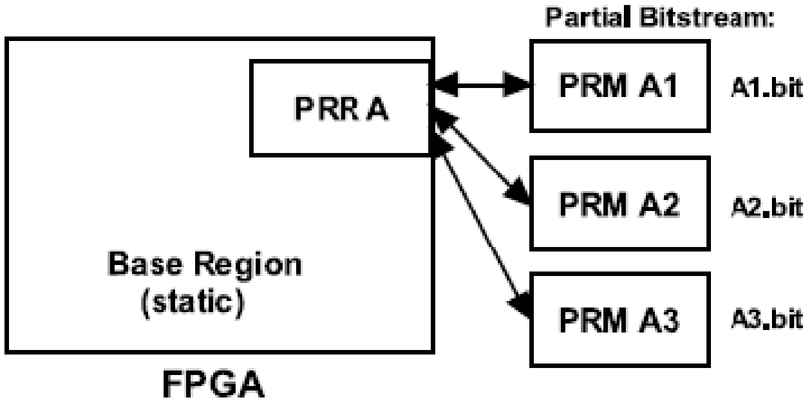
- DAQ & Trigger System
- FPGA Partial Reconfiguration (PR) Technology
- Adaptive Computing in DAQ & Trigger Systems

DAQ & Trigger System

Compute Node (CN) based DAQ & trigger system for high-energy physics experiments:

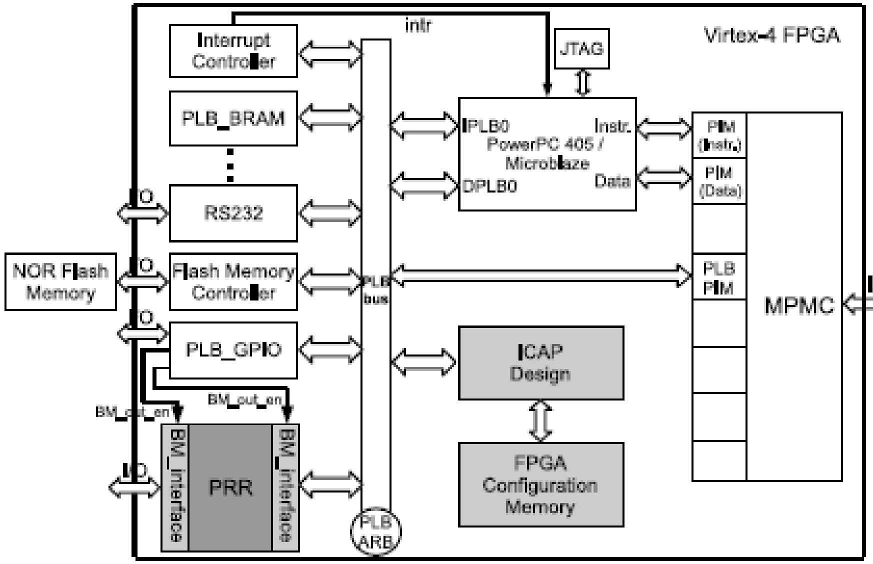
- Xilinx Virtex-4 FX60 FPGAs on board
- ATCA interconnection architecture
- Optical links & Gigabit Ethernet
- Embedded hardcore PowerPCs & Linux OS on FPGAs
- Pattern recognition algorithms implemented as hardware co-processors (RICH ring recog., MDC tracking, TOF & Shower processing., …)
- A general-purpose computation platform for multiple experiments, such as HADES, PANDA, WASA, Belle, …

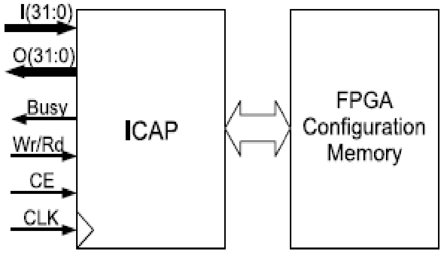
Partial Reconfiguration Technology



- PR Region (PRR) dynamically loaded with different design modules (partial bitstreams)
- Designs can be switched in the system run-time for different algorithms
- HW resources are multiplexed by different PR Modules (PRM)

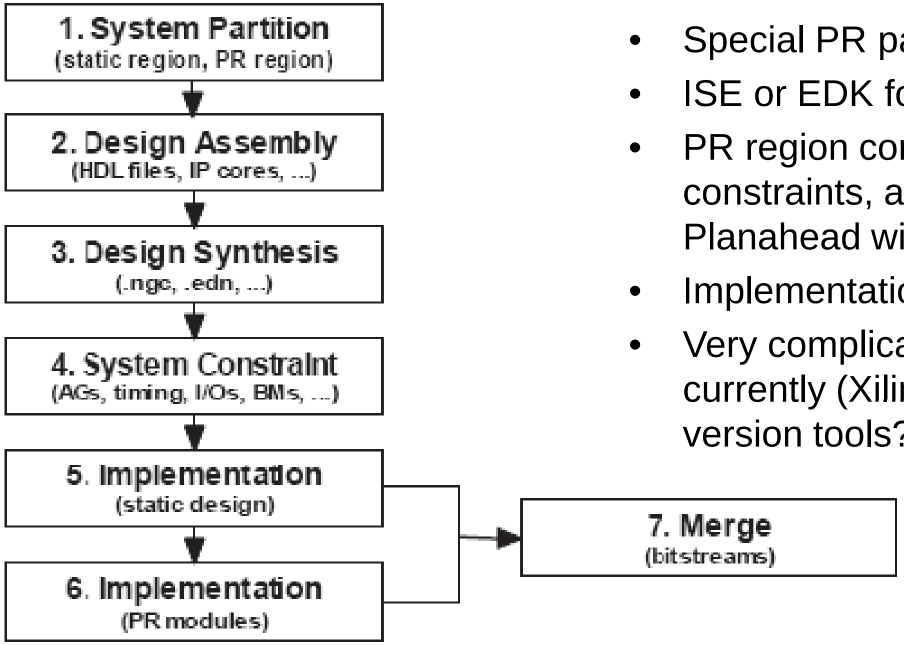
Partial Reconfigurable System





- PR region for different algo.
- Bus Macro interfaces to lock the routing between PRR & static design when implementation
- Run-time reconfiguration by ICAP (Internal Configuration Access Port)
- Dynamically load different PR Modules (PRM) by writing to ICAP

PR Design Flow



- Xilinx PR design flow
- Special PR package installation needed
- ISE or EDK for design synthesis
- PR region constraints, BM position constraints, and other constraints in Planahead with a GUI support
- Implementation in Planahead
- Very complicated design flow for PR currently (Xilinx will improve in next version tools? Or we do some work?)

Reconfiguration Speed

- Multiple ICAP designs with different interfaces (XPS_HWICAP, OPB_HWICAP, DMA_HWICAP, BRAM_HWICAP)
- Configuration time from hundreds of us to ms according to partial bitstream sizes (configuration speed of 10 MB/s ~ 370 MB/s for different ICAP designs listed above)
- Reconfiguration overhead: time needed by CPU or DMA to transport bitstream data into ICAP and FPGA configuration memory
- Theory reconfiguration bandwidth of ICAP: 400 MB/s (BRAM_HWICAP can approach to this limit, but with large BRAM resource utilization on FPGAs)

Adaptive Computing with PR

Motivation:

- Multiple pattern recognition algorithms in DAQ & trigger systems in high-energy physics experiments
- Multiple cores for each algorithm for massive parallel processing
- Computation steps distributed on FPGAs
- Difficult to manage and modify the large system (many FPGAs, many algorithms, many cores, different FPGA bitstreams, long design synthesis & implementation time, …)
- Also different computation features for algorithms (computation-bounded, memory-bounded, …)
- Traditionally all partitions are considered by designers during system development process, rather than dynamically and online.

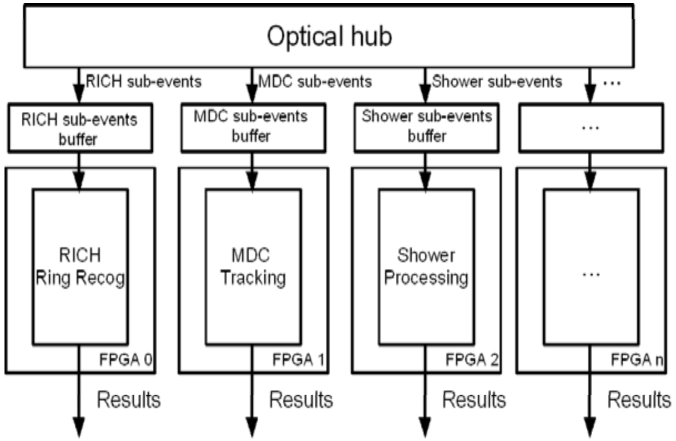
Adaptive Computing with PR

One promising solution: adaptive computing

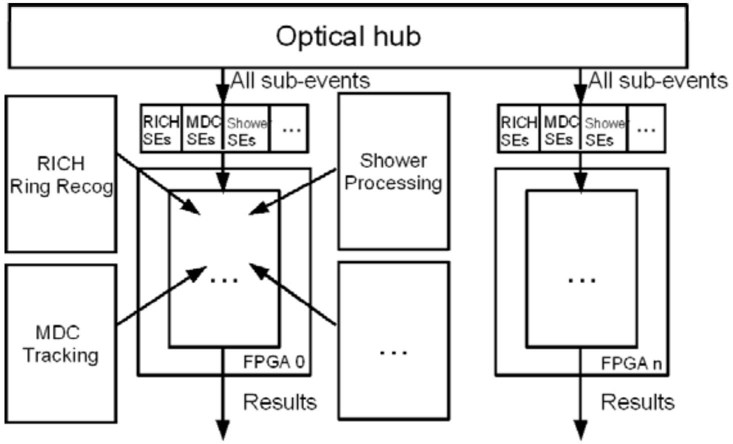
- Algorithm cores designed as PR modules
- Modules can be adaptively loaded during experiments, according to external factors (workload, sub-event types, …)
- Uniform DAQ & trigger design to interface with optical hubs, which delivers all kinds of sub-events
- Performance improvement due to the balance of computation and memory accesses, as well as more efficient utilization of FPGA resources? (studies needed)
- Other merits?... (to be explored)

One Example for DAQ & Trigger

Traditional non-PR design:



PR design for adaptive computing:



- Uniform design in adaptive computing – easy to maintain system designs
- No data distribution requirements for optical hubs (all kinds of sub-events fed into all FPGAs)
- Balanced computing and more efficient FPGA resource utilization

On-going and Outlook

Currently

- PR design flow has been studied.
- PR speed of ICAP designs has been investigated and improved.
- Basic concept of adaptive computing in DAQ & trigger system has been considered and proposed.

In future

- More systematic design flow will be investigated for adaptive computing in DAQ & trigger systems.
- Performance study and optimization will be done to compare with the traditional static designs.

Thanks for your attention!