**1) The FASTER Project**

**Goal:** Reducing the complexity of Dynamic Partial Reconfiguration

- **RTSM** fed with **static** and **dynamic inputs**; 
  - **static**: from XML with scheduling alternatives to reconfigure the FPGA 
  - **dynamic**: updated with FPGA runtime status

**Run-time Reconfiguration Management**

Primarily focused on **partial dynamic reconfiguration** with two options:

- **a)** Region-based: large device portions, precompiled circuits
- **b)** Micro-reconfiguration: small circuits specialized at runtime based on dynamic inputs

**2) RTSM Reference System Model**

- **Application (parallel specification: tasks, threads) → FPGA**
- **CPU**
- **Parallel, Reconfigurable Platform**
- **SW/HW Partitioning & Mapping Front-end**
- **HW Task Graph**
- **Identification/Optimization of RC portions**
- **Vendor FPGA synthesis & PR tools**
- **HW Task Graph Annotations** (performance, power, etc)
- **Platform Description**
- **Run-Time SW Scheduler**
- **Static data (Task Graph, Run-Time SW Scheduler)**
- **Dynamic data (schedule meta data)**

**3) RTSM Functionality**

- **Place**: Find the best location on the FPGA
- **Scheduler**: Find best loading, executing times
- **Translator**: Transform technology independent data to platform specifics
- **Loader**: Communicate with configuration ports

**4) RTSM Architectural Interface**

**Technology Independent Bitstream Format with Micro reconfiguration support**

- **BS Length**
- **BS Address**
- **Input Data Address**
- **Output Data Address**
- **Input Data Format**
- **Output Data Format**
- **Task Parameters**
- **Task Configuration Microcode**

**Configuration agnostic ISA extension with:**

- Full support for loosely coupled accelerators
- Support for micro-reconfiguration and for two level nested region/micro-reconfiguration
- Reconfiguration in 2 logical phases: **SET EXECUTE address**
- Total of 5 new instructions: **P-SET, C-SET, EXECUTE, BREAK, SET PREFETCH**
- Native Partial Reconfiguration Support
  - **P-SET (partial set):** configure common parts of multiple (frequently used) functions
  - **C-SET (complete set):** configure the remaining blocks (not covered by P-SET) to complete the functionality
- Support for bitstream prefetching: **SET PREFETCH**

**5) RTSM Prototype**

**RTSM prototyped on a desktop PC with:**

- Intel Core i7-950, 3.06GHz
- CentOS Linux 6.4 64-bit
- 6GB DDR3@1600MHz
- One reconfigurable area
- Host CPU controls FPGA reconfiguration
- Bitstreams stored in HDD
- Bitstreams cached in host RAM

**6) Results and Future Directions**

**A)** Xilinx University Program XUPV5

- **Virtex 5-LX110T**
- Two XC6SLX32 Platform Flash PROMs 32MB each
- 256MB DDR2
- PCIe v1 x1, 2Gbps bandwidth

**B)** HiTechGlobal HTG-V5-PCIe-330

- **Virtex 5-LX330T**
- 4 MB Flash Memory
- 2 GB DDR2
- PCIe v1 x4, 8Gbps bandwidth

<table>
<thead>
<tr>
<th>XUPV5 on PCIe v3 (x1)</th>
<th>HTG-V5 on PCIe v1 (x4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A) Transfer Rate</td>
<td>1.5 Gbps</td>
</tr>
<tr>
<td>B)</td>
<td>4 Gbps</td>
</tr>
<tr>
<td>Reconfiguration Rate</td>
<td>0.8 Gbps</td>
</tr>
<tr>
<td></td>
<td>3.5 Gbps</td>
</tr>
</tbody>
</table>

Results show that the transition to faster PCIe mitigated the PCIe bottleneck:

- Increased transfer by 2.7 times (CPU ↔ FPGA)
- Increased reconfiguration rate by 4.4 times (CPU ↔ p FPGA)

**Future (ongoing) work:** Maxeler MaxWorkstation

- 32x dataflow engine
- **Virtex 6 5X475T FPGA, 24GB memory**
- Intel i7 2600x@2.8GHz, 16GB RAM