TPC Electronics

FPGAs & Dynamic (Re)Configuration – ongoing work

Gerd Tröger
Kirchoff-Institut für Physik, Univ. Heidelberg
troeger@kip.uni-heidelberg.de

Contents

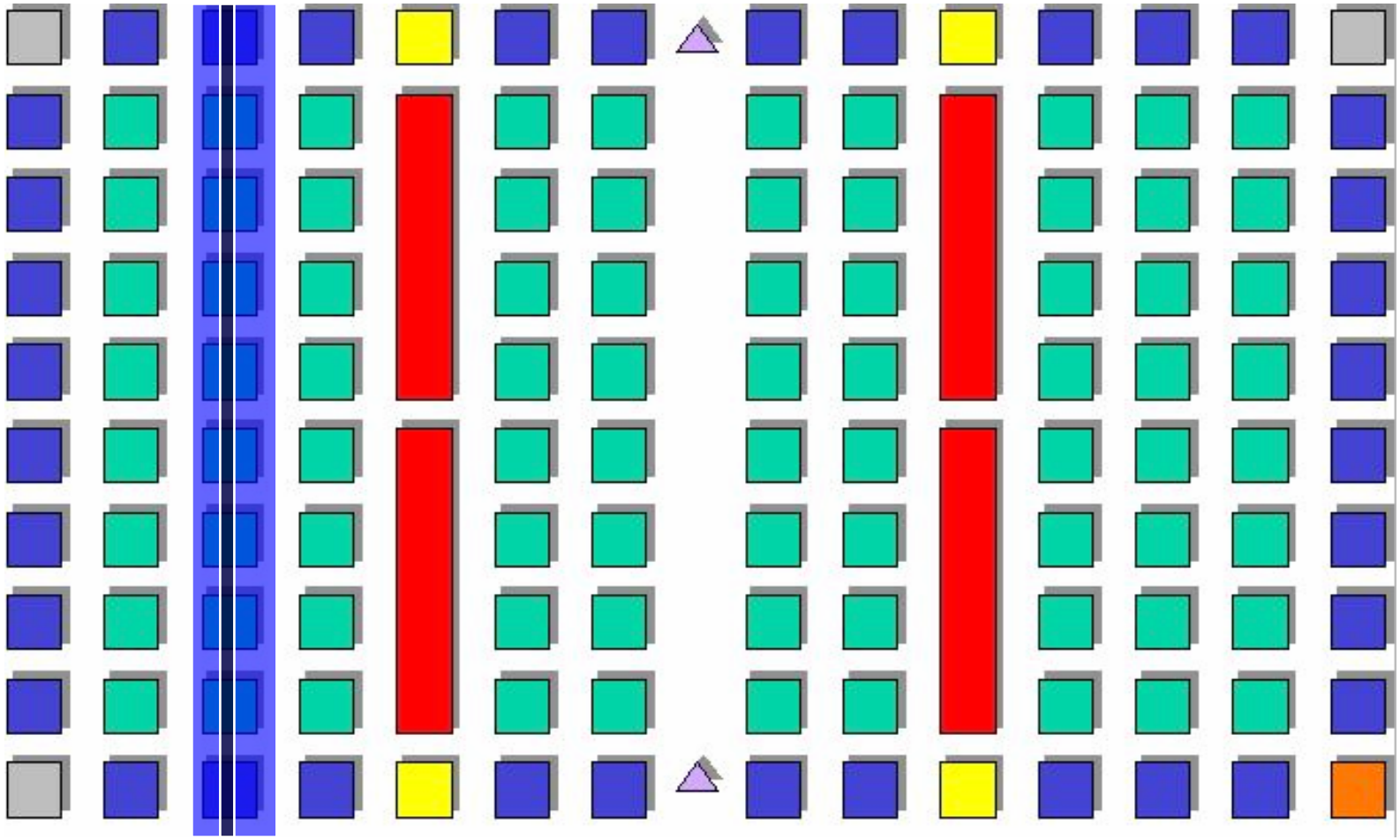
- **Past Work**
  - **Virtex-II Pro Irradiation Tests**
  - **SelectMAP Controller for DCS Card**
- **Upcoming Work**
  - **FPGA Debugging**
  - **Dyn.Reconfig.: Modules – Generic Approach**

SelectMAP Controller for DCS Card

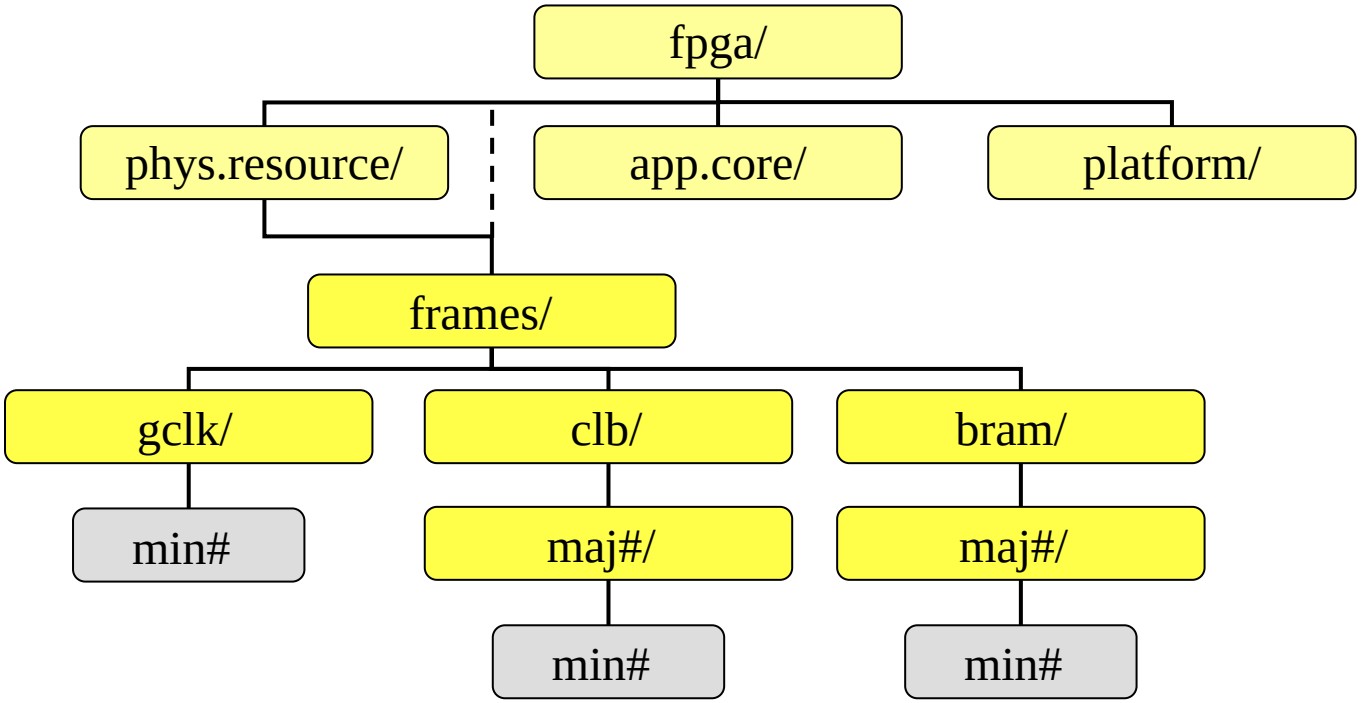
SelectMAP Controller for DCS Card

- **AHB SelectMAP controller**
  - **Altera/ARM can (re)configure Xilinx Virtex**
  - **Fully implented in hardware, i.e. fast**
- **Basic Linux drivers**
- **FPGA Virtual Filesystem (Linux)**

FPGA Configuration Space



FPGA Virtual File System



Hardware Debugging

FPGA / Hardware Debugging

- **Common Approaches:**
  - **Software Simulation**
  - **Hardware Supported Simulation (FPGA farms)**
  - **Modified Designs in FPGA**
- **Either too slow, or too expensive, or too limited capabilities, or …**

Minimum Invasive FPGA Debugging

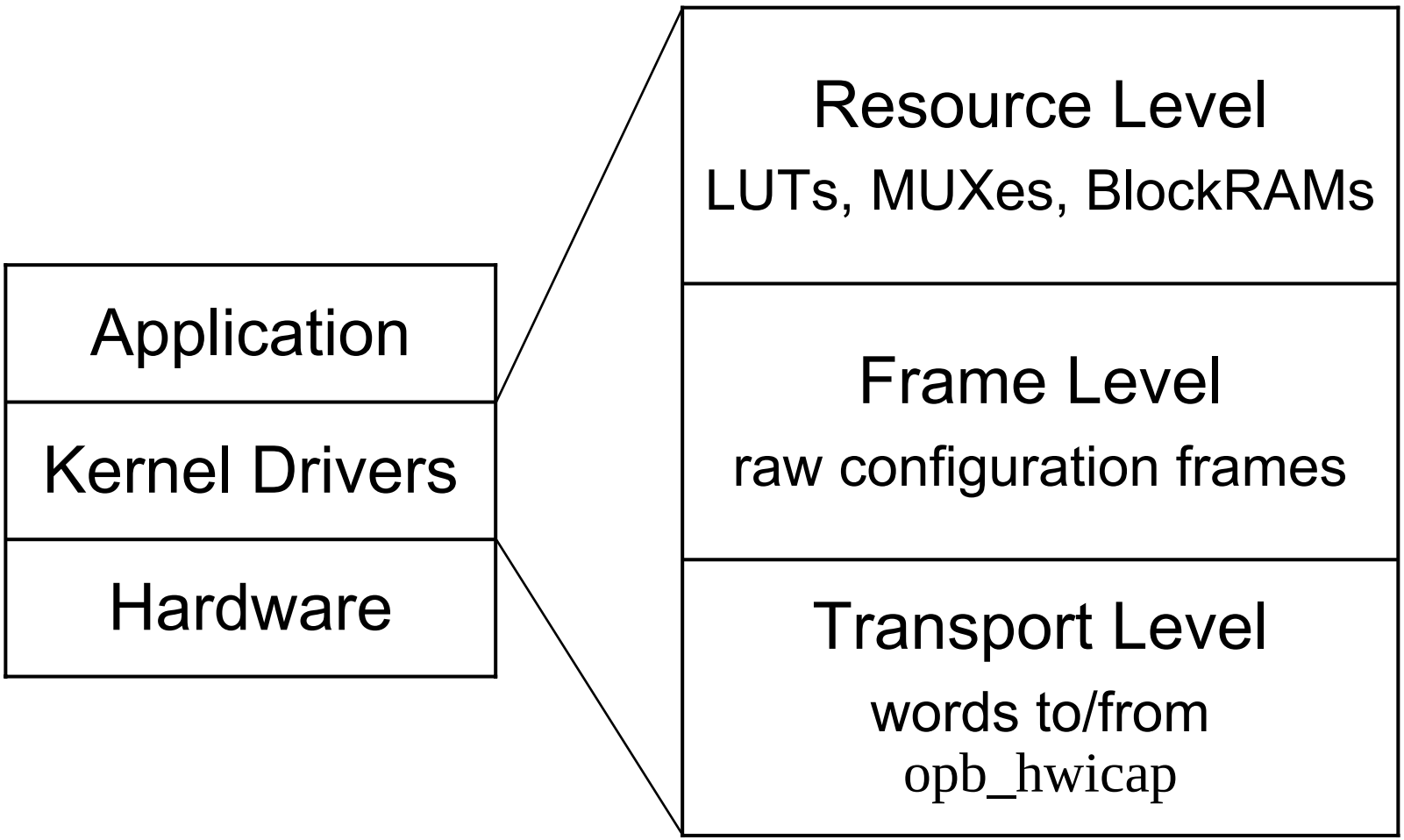
- **Wishlist:**
  - **Unmodified (Original) Design in FPGA**
  - **Software-Debugger-Like Features**
    - » **Breakpoints (not w/o design modifications)**
    - » **Single-Stepping**
    - » **Memory Inspect & Edit**

‘Memory’ Inspecting & Editing

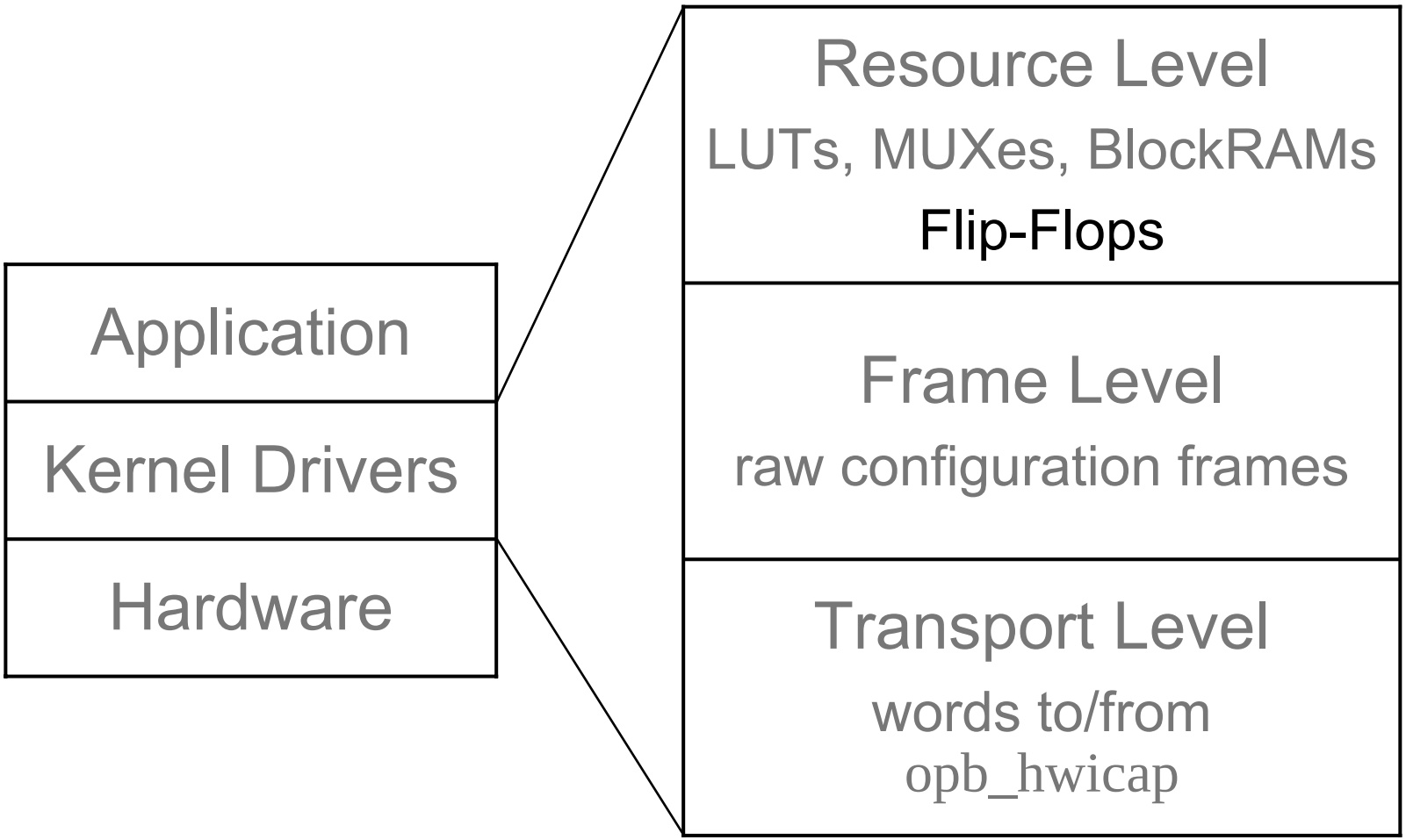
- **Reading & writing of memories**
- **Memories =**

  Flip-Flops, LUT-RAMs, BlockRAMs

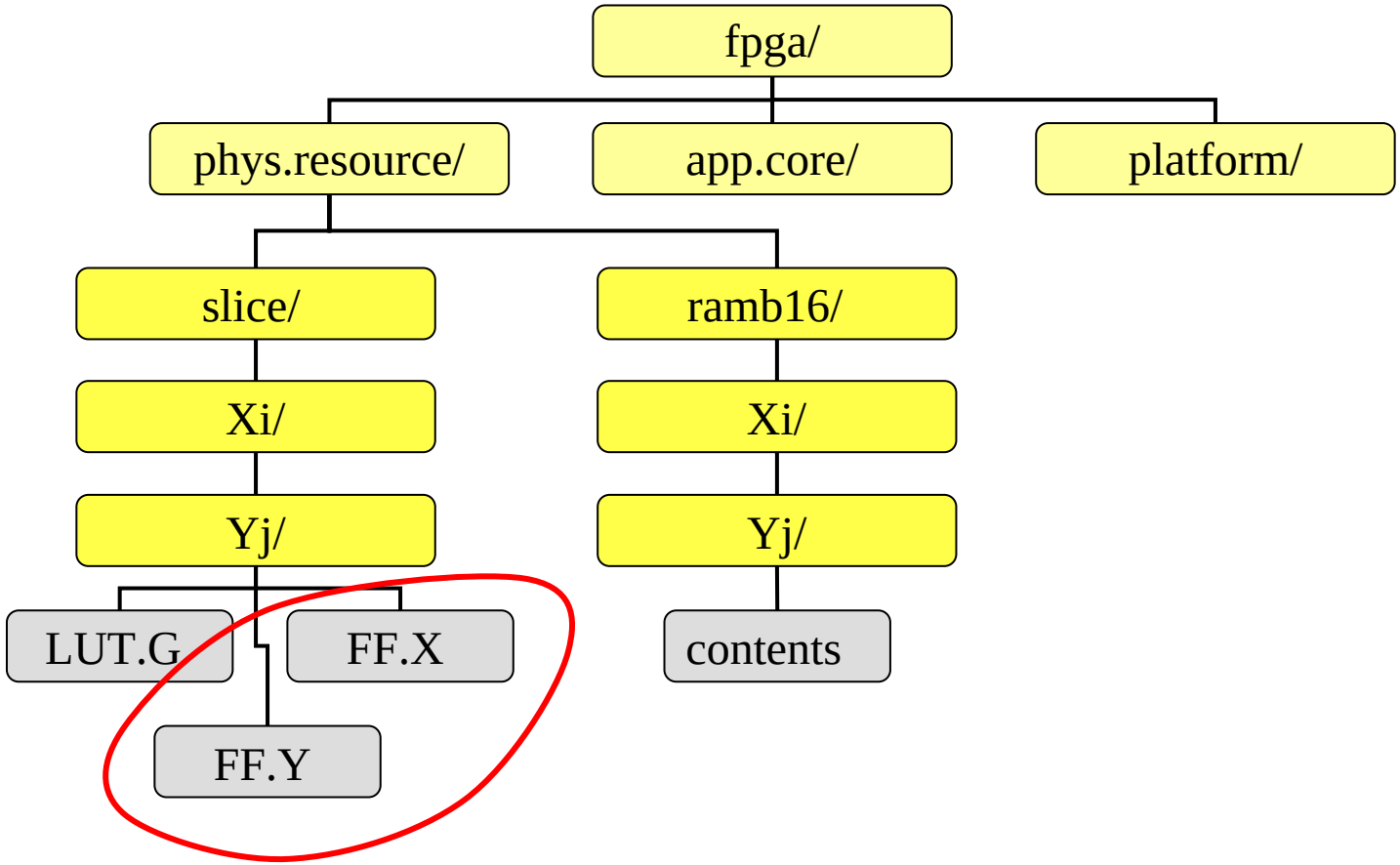
ICAP / SelectMAP Linux Driver



ICAP / SelectMAP Linux Driver



FPGA Virtual File System



FPGA VFS Examples: App.Core Configuration

```
<core name="test">
  <resource name="x67y60.lut.g" type="bits">
    <element data="SLICE_X67Y60:2:24"/> […]
    <element data="SLICE_X67Y60:2:39"/>
  </resource>
</core>
```

Automated FF Extraction

- **after synthesis, place & route**
- **extract name and position of all flip-flops**
  → meta-data for FPGA VFS

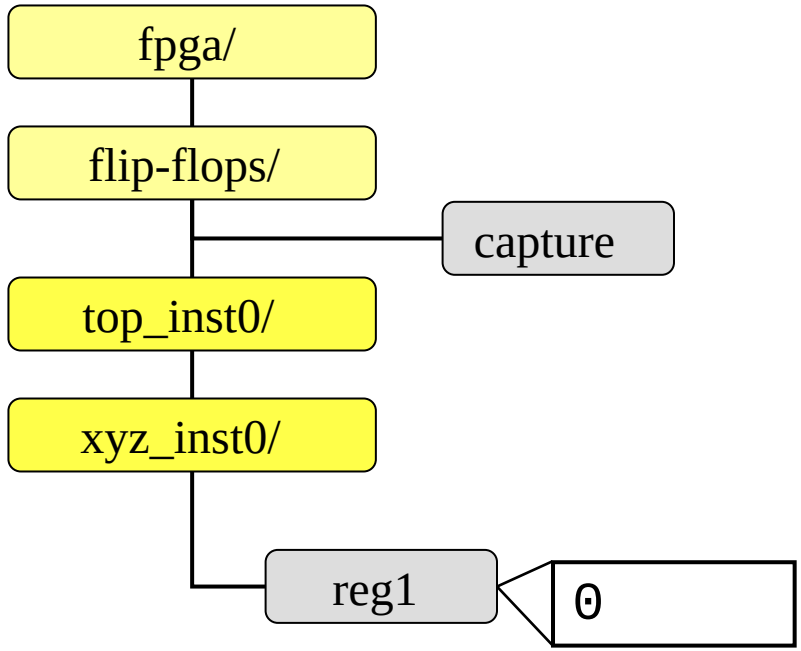
Automated FF Extraction

source.v:

```
module xyz (ports);

reg          reg1;


always @(posedge clock)
begin
  …
end;
```



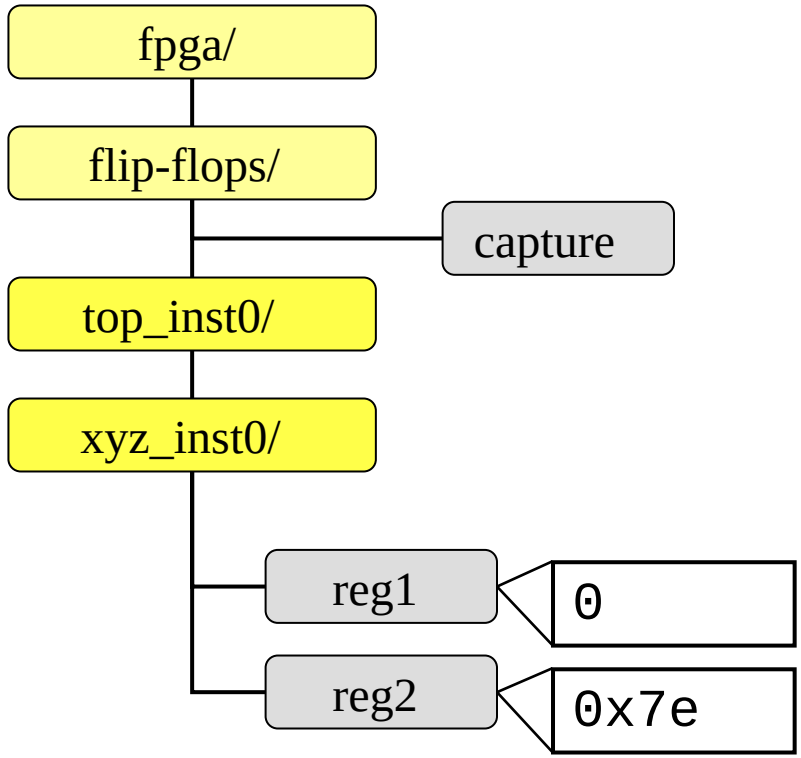
Automated FF Extraction

source.v:

```
module xyz (ports);

reg          reg1;
reg [0:7]    reg2;

always @(posedge clock)
begin
  …
end;
```



Hardware Single-Stepping

- Step by step execution of a design
- Like cycle-accurate simulation
- But: actual design on the real boards
- Requires: clock control

Hardware Single-Stepping

- Global Clock Buffers: Multiplexers
- can be used to enable/disable clock

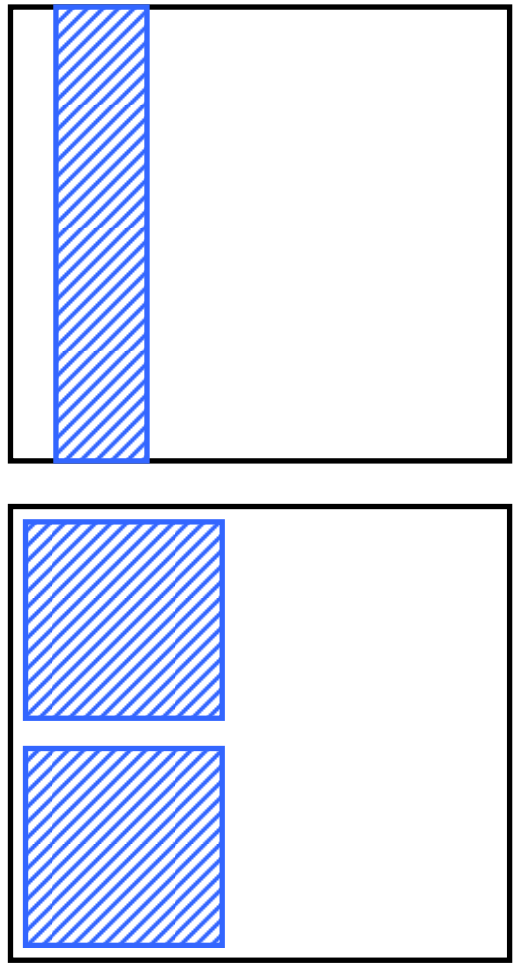
- feed mux.select from a ‘constant’ LUT
- reconfigure LUT to switch between clocks

  → single-stepping

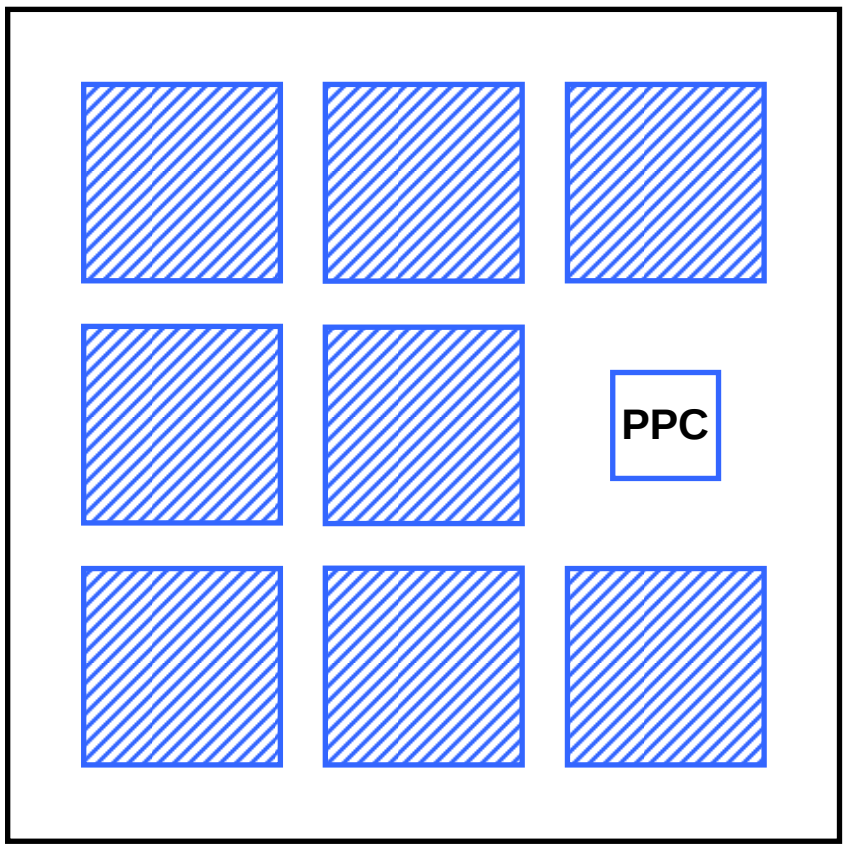
Dynamic Reconfiguration
Modules – Generic Approach

Reconfigurable Slots

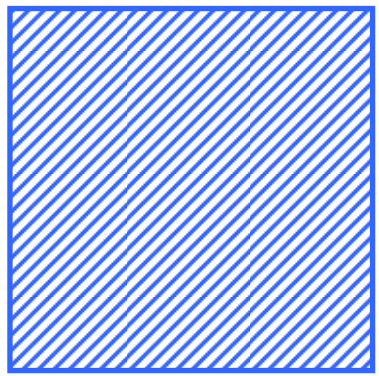
- **Partial reconfiguration:**
  - **Frame based, full columns only**
- **Reconfigurable slots:**
  - **‘Arbitrarily’ shaped areas**
  - **Granularity: CLB**
  - **Relocateable**
- **Prototype in perl**
  - **Using VFS Frame View**



Reconfigurable Slots – Improved

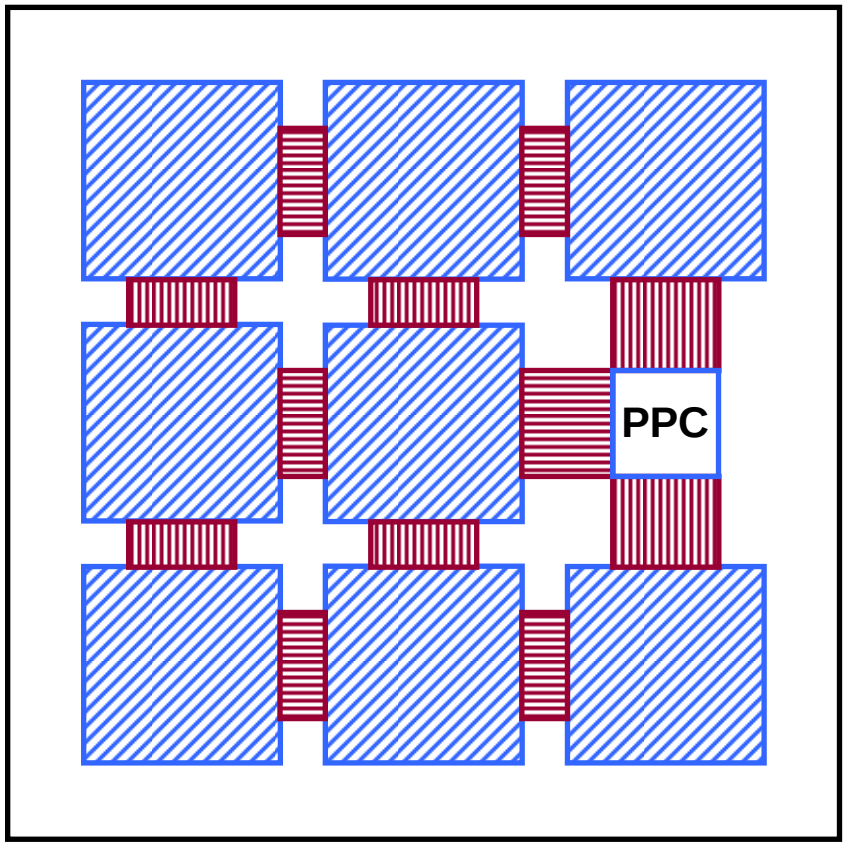


FPGA with module slots

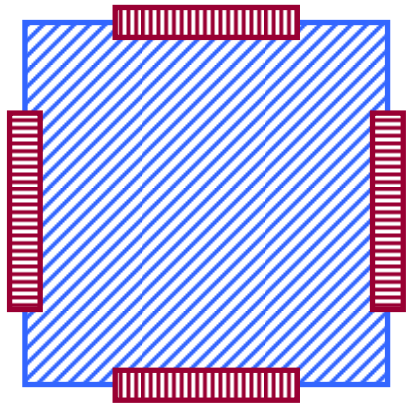


single module

Reconfigurable Slots – Improved

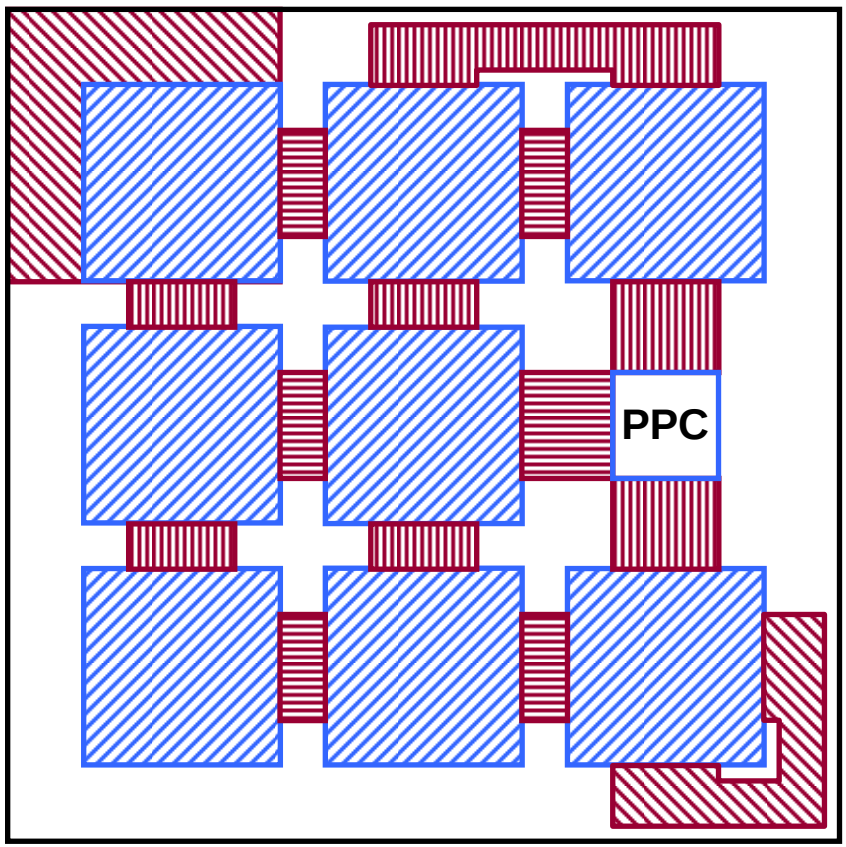


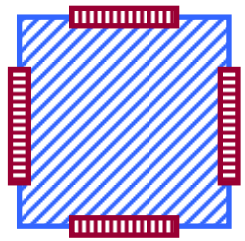
FPGA with module slots



single module

Reconfigurable Slots – Improved





- **Interconnect:**
  - **direct**
  - **buffered (flip-flops)**
  - **buffered (BRAM FIFO)**
- **Border:**
  - **external I/O and/or**
  - **flexible topologies**

Reconfigurable Slots – Goals

- **Configuration Level Compatibility**
  - **freely relocatable modules**
- **Generic Interconnect**
  - **simple data bus with flow-control**
  - **several options: direct, buffered, FIFO, …?**
- **Integration in FPGA VFS**
  - **including clock management etc.**

Reconfigurable Slots – Application

- **Possible application for TPC / ALICE:**
  Flexible Data Filter/Trigger System

EOF