On-chip Context Save and Restore of Hardware Tasks on Partially Reconfigurable FPGAs

Aurelio Morales-Villanueva and Ann Gordon-Ross

CHREC NSF Center for High-Performance Reconfigurable Computing

UF UNIVERSITY of FLORIDA

Introduction

Goals

- Develop services that leverage partial reconfiguration (PR) for high performance embedded reconfigurable computing (RC) systems
- Exploit services to give the user control over application development

Motivations

- PR enables area and power savings
- PR-capable FPGAs enable preemption/resumption of hardware tasks (i.e., PR modules (PRMs)) in PR regions (PRRs) without losing tasks' execution state
- PR-capable FPGAs enable hardware multitasking over shared resources

Approach

- Leverage PR capabilities of FPGAs
- Implement services with portability across different FPGA architectures

Accomplishments

- Implemented on-chip context save and restore (CSR) software to save and restore PRM's execution context

On-chip Context Save and Restore (CSR)

Overview

- CSR provides a mechanism to capture execution state of preempted PRMs and resume PRM execution in the same PRR

Approach

- Context save (CS) and context restore (CR) via ICAP
  - CS: Checkpoint PRM's current execution state (i.e., context)
  - CR: Restore checkpointed PRM execution state
- Implement as a C program running on MicroBlaze

Benefits

- On-chip, no external device required
- Software on MicroBlaze, no custom hardware needed
- Avoids lengthy re-execution time of PRMs
- No tool flow changes needed

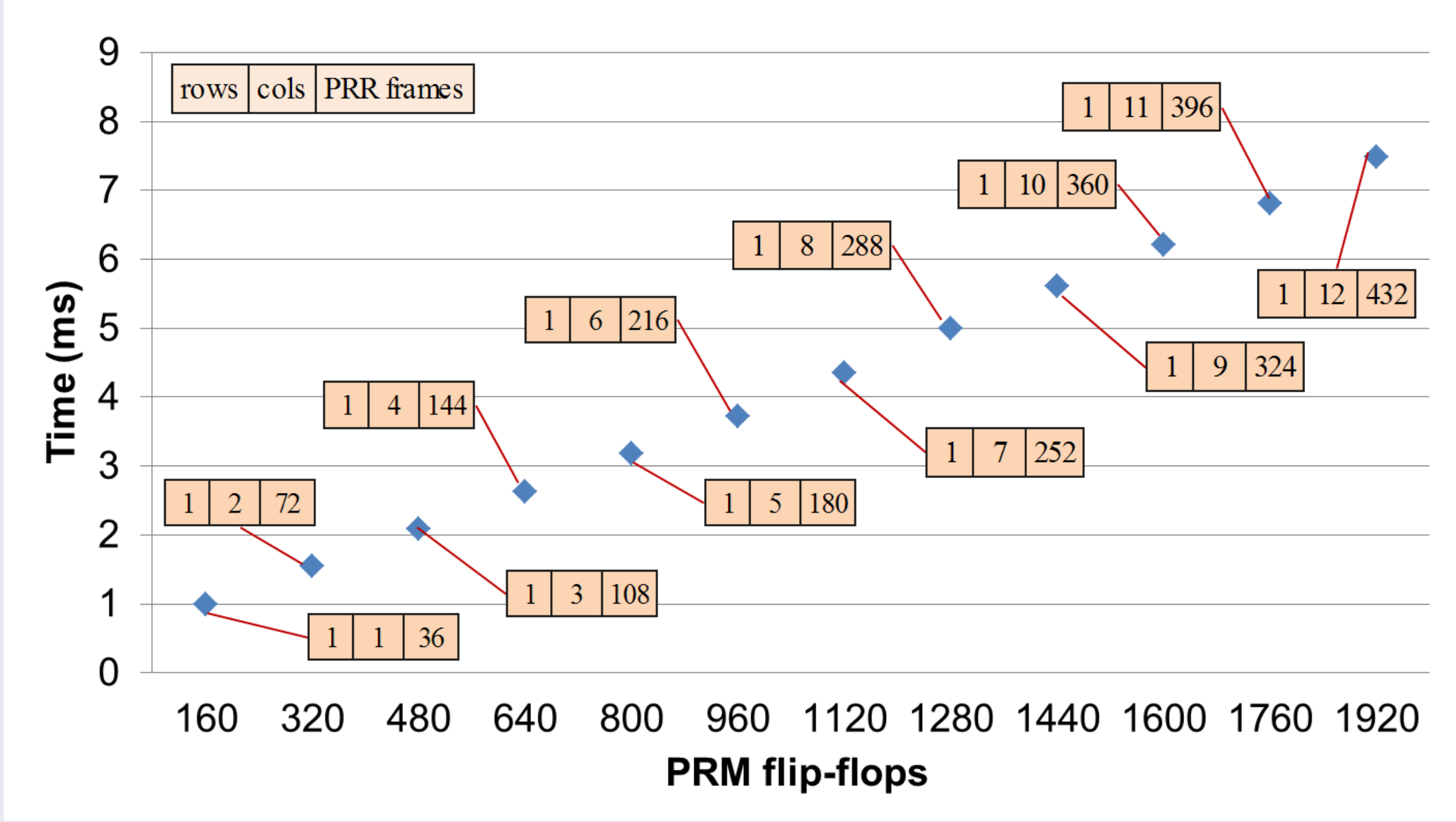
PRM	– Partially Reconfigurable Module
PRR	– Partially Reconfigurable Region
CLB	– Configurable Logic Block
ICAP	– Internal Configuration Access Port
UART	– Universal Asynchronous Receiver/Transmitter
FSL	– Fast Simplex Link
GPIO	– General Purpose Input/Output

Experiments

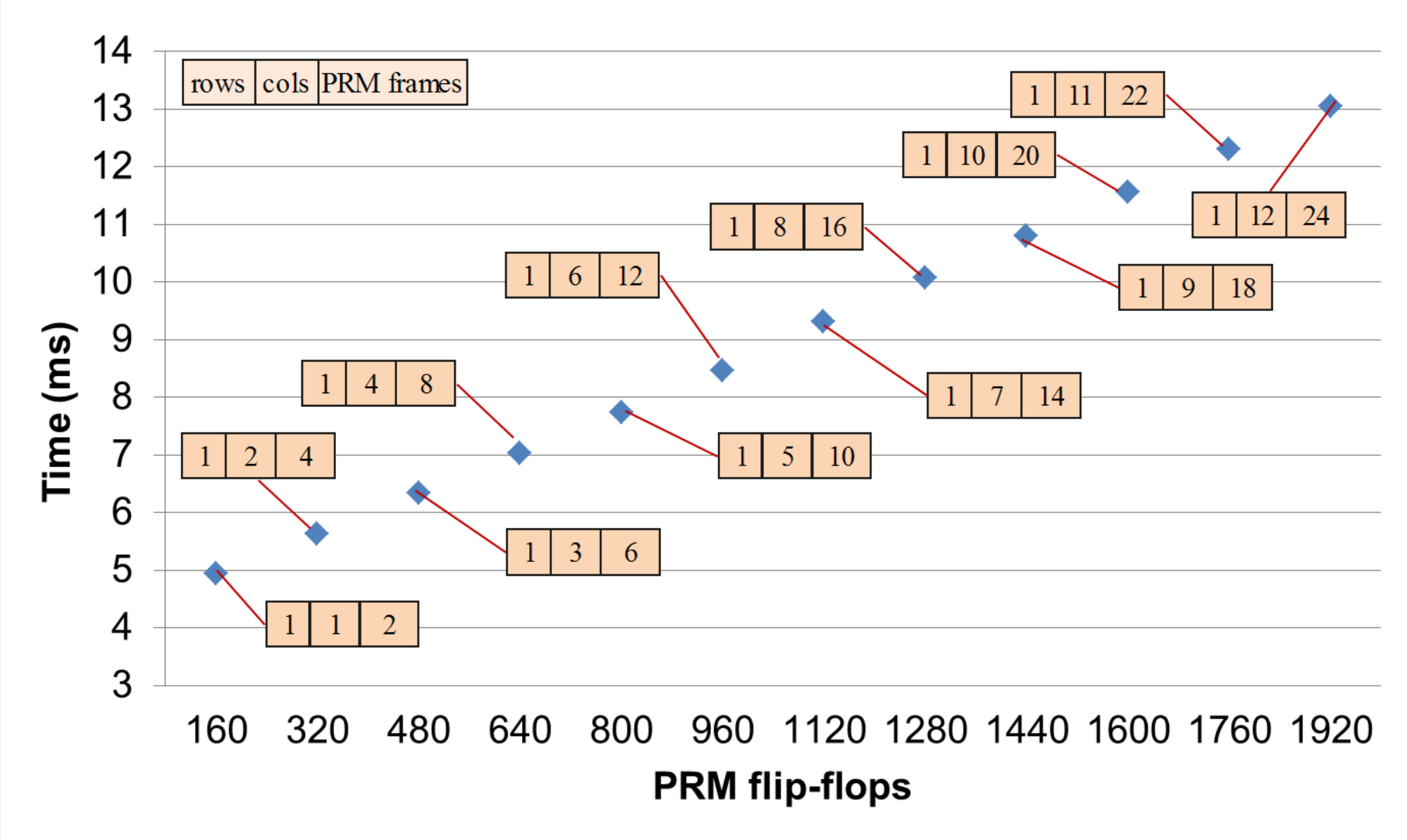
- Static region: MicroBlaze, RS232 UART, Ethernet interface, FSLs, ICAP, GPIOs, DDR2 SDRAM, Compact Flash interface
- PRRs: Each PRR implements two or more PRMs
- Testbed: Virtex5 LX110T, 100 MHz, 1 PRR, OpenSPARC board, Linux OS

Experimental Results

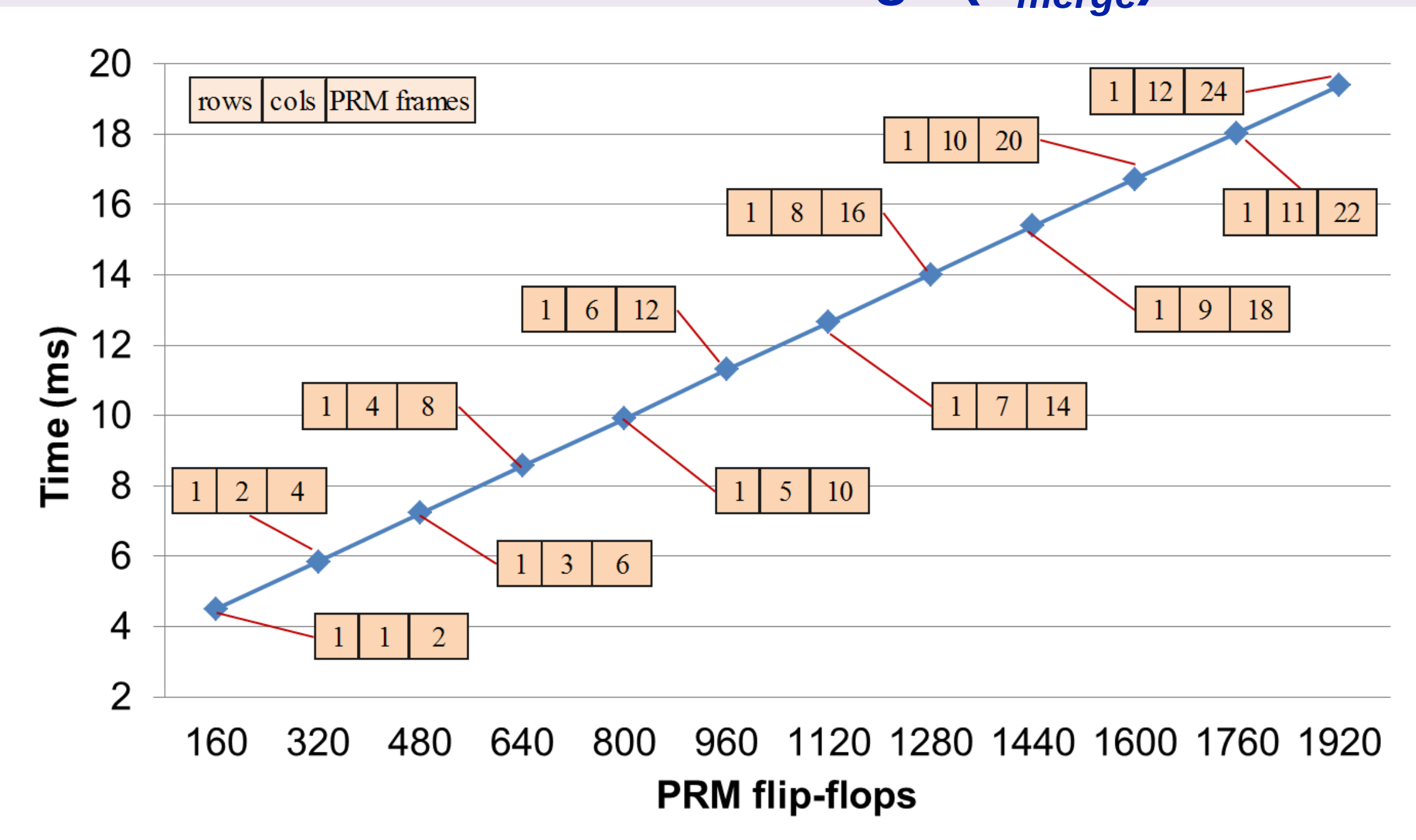
Execution time to reconfigure PRR ($T_{reconfigure\_PRR}$)



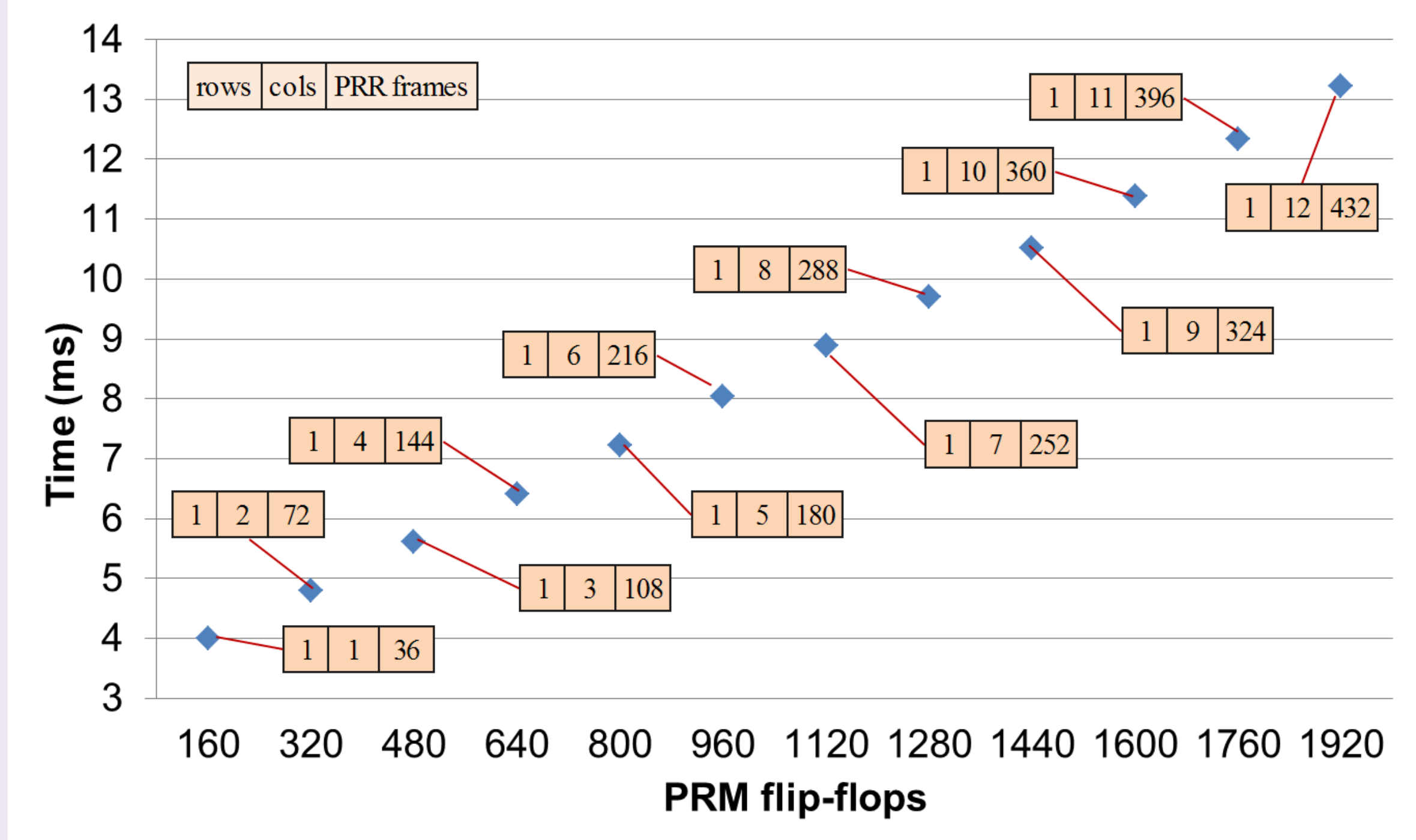
Execution time for Context Save (T_{CS})

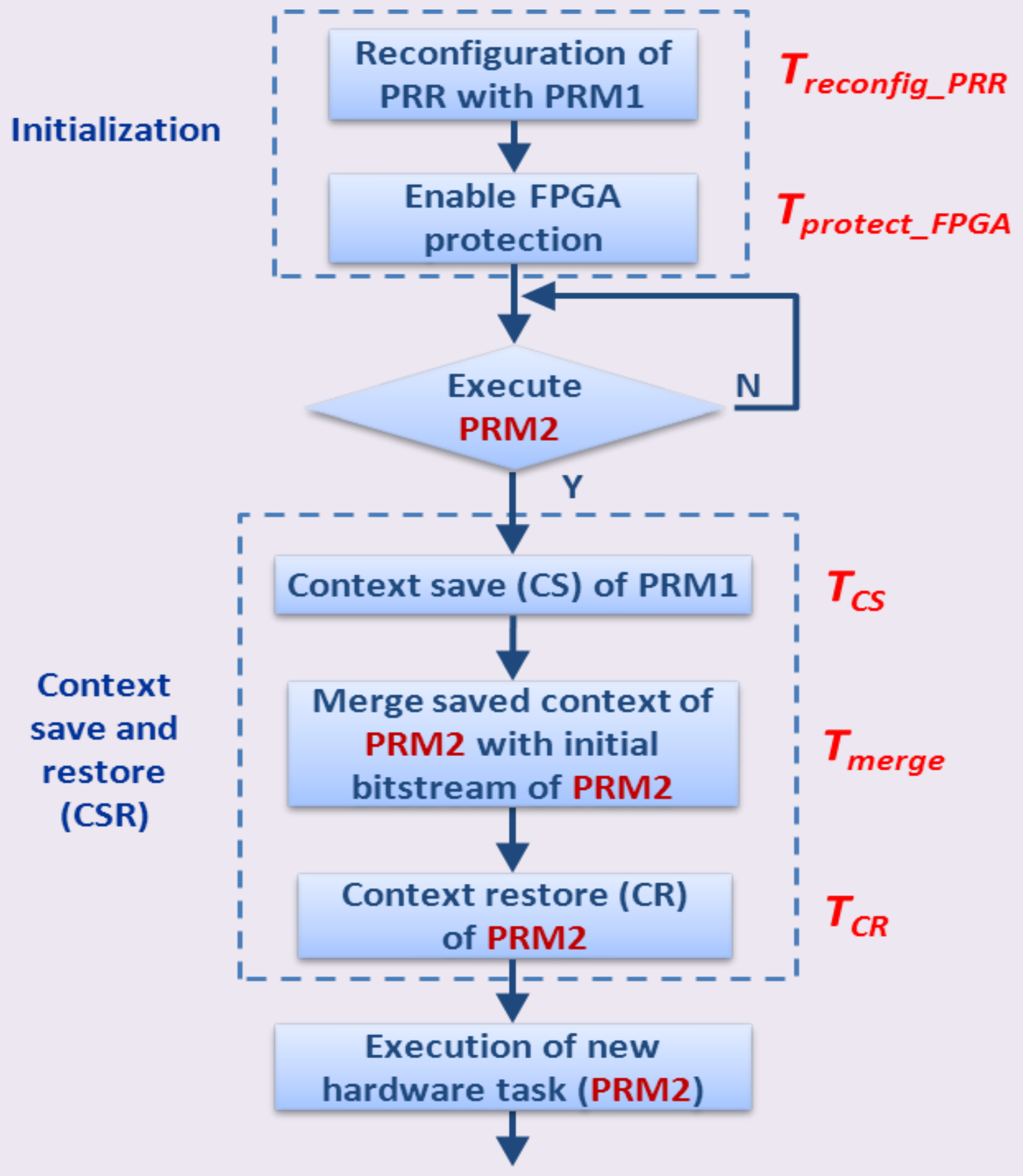


Execution time for Merge (T_{merge})



Execution time for Context Restore (T_{CR})




Initialization
Reconfiguration of PRR with PRM1
$T_{reconfig\_PRR}$
Enable FPGA protection
$T_{protect\_FPGA}$
Execute PRM2
N
Y
Context save and restore (CSR)
Context save (CS) of PRM1
T_{CS}
Merge saved context of PRM2 with initial bitstream of PRM2
T_{merge}
Context restore (CR) of PRM2
T_{CR}
Execution of new hardware task (PRM2)

Context save and restore (CSR) flowchart

CSR flowchart details

- Example: 1 PRR, 2 PRMs per PRR
  - PRR is CLBs only
  - PRR is 1 row, many columns
- Protection
  - Entire FPGA before first CS
  - Only on PRR after CR
- Unprotection
  - Only on PRR before CS and CR

CSR experimental results

- $T_{reconfig\_PRR}$ shows linear growth rate
  - Depends on the total number of frames in PRR
- T_{CS} shows linear growth rate
  - Depends on the number of frames that contain CLB flip-flops
  - Includes unprotection and re-protection of PRR
- T_{merge} shows linear growth rate
  - Depends on the number of flip-flops used in the PRR
- T_{CR} shows linear growth rate
  - Depends on the total number of frames in the PRR
  - Includes unprotection and re-protection of the PRR