



# Reconfigurable Systems



Mikael Olausson

Computer Engineering  
Department of Electrical Engineering  
Linköping University

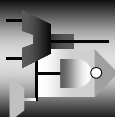
26/10/2001 Reconfigurable Systems 1



## Outline

- Temporal Partitioning
- Temporal Partitioning with Partial Reconfiguration
- Embedded Reconfigurable Architectures
- Conclusions

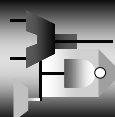
26/10/2001 Reconfigurable Systems 2



## Temporal Partitioning

- M. Kaul, R. Vemuri, "Temporal Partitioning Combined with Design Space Exploration for Latency Minimization of Run-Time Reconfigured Designs", Proc. DATE 1999
- Temporal Configuration
- Application partitioning

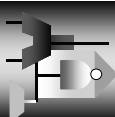
26/10/2001 Reconfigurable Systems 3



## Alternatives

- Many different implementations
  - Area
  - Latency
- Integrate partitioning with synthesis
- Iterative process
  - Lowest latency that meets area

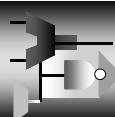
26/10/2001 Reconfigurable Systems 4



## Partitioning Levels

- Behavior level
- Register Transfer Level
- Gate level

26/10/2001 Reconfigurable Systems 5



## Design Points

- Different implementations of the same task
  - Time-Area tradeoff
  - Serial vs Parallel
- Too many design points?
  - Candidate design points

26/10/2001 Reconfigurable Systems 6

**any of e**  
**artitionings**

- Spatial Partitioning
  - Increase partitioning
  - Consumes more area
  - Parallel processing => Less latency

26/10/2001 Reconfigurable Systems 7

**any of e**  
**artitionings cont**

- Temporal Partitioning
  - Increase partitions
  - Increase the available area
- Will Latency decrease?
  - Heavily dependent on the reconfiguration times

26/10/2001 Reconfigurable Systems 8

**artitioning**

1. Map tasks to partitions
2. Map each partition to several design points
3. Explore multiple implementations of the design point

26/10/2001 Reconfigurable Systems 9

**n ut to lgorit m**

- Behavior specification (Task graph)
  - Tasks
  - Communication between
- Target Architecture
  - Area
  - Memory size
  - Configuration times

26/10/2001 Reconfigurable Systems 10

**e lgorit m**

- Find the constraints
  - Minimum number of partitions, lower bound  $N_{min}^l$
  - Minimum number of partitions, upper bound  $N_{min}^u$
  - Worst case latency  $D_{max}$
  - Best case latency  $D_{min}$

26/10/2001 Reconfigurable Systems 11

**lgorit m Ste s**

1. Find one solution for the constraints
2. Tighten the latency constraints
3. Increase the partition size and start over

26/10/2001 Reconfigurable Systems 12

## Search limits

- New  $D_{max} = (D_{max} + D_{min}) / 2$
- Stop when  $D_{max} - D_{min} < \delta$  or when no new solutions are found time limit
- Start and stop parameters for the partitioning search  $\alpha$  and  $\gamma$
- When reconfiguration time is large, set  $\alpha = \gamma = 0$

26/10/2001 Reconfigurable Systems 13

## Examples

- 4x4 DCT with low reconf. Time 10ns

| N | I | Bounds    |           | Result |   |
|---|---|-----------|-----------|--------|---|
|   |   | $D_{max}$ | $D_{min}$ | $D_a$  | T |
|   |   |           |           |        |   |

26/10/2001 Reconfigurable Systems 14

## Examples

- 4x4 DCT with high reconf. Time 30ms

| N | I | Bounds    |           | Result |     |
|---|---|-----------|-----------|--------|-----|
|   |   | $D_{max}$ | $D_{min}$ | $D_a$  | T   |
| 8 | 1 | 25,440    | 795       | Inf    | 300 |

26/10/2001 Reconfigurable Systems 15

## Partial Reconfiguration

- S. Ganesan, R. Vemuri, "An Integrated Temporal Partitioning and Partial Reconfiguration Technique for Design Latency Improvement", Proc. DATE 2000.
- One part executing, one part reconfiguring

26/10/2001 Reconfigurable Systems 16

## Partial Recon cont

26/10/2001 Reconfigurable Systems 17

## asic once t

- For maximum overlap:
  - $Exe(Tp_i)$  comparable to  $Rec(Tp_{i+1})$
  - Or  $Exe(Tp_i) \geq Rec(Tp_{i+1})$

26/10/2001 Reconfigurable Systems 18

## Syntax

- Input behaviour specification in C or VHDL
- Generate a Control Data Flow Graph(CDFG)
- Partitioner + area estimator => Temporal segments
- High-level synthesis => RTL

26/10/2001 Reconfigurable Systems 19

## Input specification

Behaviour Block Intermediate Format (BBIF)

26/10/2001 Reconfigurable Systems 20

## Target architecture

- Xilinx 6200 FPGA

Switch between execution and reconfiguration

26/10/2001 Reconfigurable Systems 21

## Loop partitioning

- Entire loop in one partition
- Why?
  - Easy partitioning
  - Execution time maximum overlapped
- If the loop don't fit?
  - Report a failure
  - Use the whole device(the adopted one)

26/10/2001 Reconfigurable Systems 22

## Conditional execution

- We have to wait for the outcome of the conditional executing
- Conditional in one, branches in the other
- If this fails
  - Host polling

26/10/2001 Reconfigurable Systems 23

## Local processing

- Gives high execution times
- One configuration for many inputs
  - Filters
  - FFT
- Works only with no dependencies between inputs

26/10/2001 Reconfigurable Systems 24

## Result

| Design | Method | #TP | #Inp blocks | Rec. Time (us) | Exe time (us) | Throughput (ms) | % rec. | Speed up vs full |
|--------|--------|-----|-------------|----------------|---------------|-----------------|--------|------------------|
| TLC    | PR     | 1   | 1           | 86             | 0.9           | 0.087           | 98.9   | 1x               |
|        | FR     | 1   | 1           | 86             | 0.9           | 0.087           | 98.9   |                  |

26/10/2001 Reconfigurable Systems 25

## Questions

- ? What is required for good performance?
- ? Would more partitions be better?
- ? Can parallel processing increase the performance?

26/10/2001 Reconfigurable Systems 26

## Reconfigurable Architectures

- Yanbing Li, et al., "Hardware-Software Codesign of Embedded Reconfigurable Architectures", Proc. DAC 2000.
- Speed up execution with FPGA

26/10/2001 Reconfigurable Systems 27

## Target Architecture

● Target Architecture

```

graph LR
    Mem[Mem] <--> CPU[CPU]
    Mem <--> FPGA[FPGA]
    CPU <--> FPGA
  
```

26/10/2001 Reconfigurable Systems 28

## Implementation


- HW/SW partitioner
- From system-level described in C
- Loop and Basic block level
- Two dimensional partitioning
  - Spatial
  - Temporal

26/10/2001 Reconfigurable Systems 29

## Implementation cont.

- Search for candidate loops for implementation in HW
- 1 SW loop vs. 1 or more HW loops
- Search for Instruction Level Paralellism


26/10/2001 Reconfigurable Systems 30



## Key issues by partitioning

- Dynamic reconfiguration costs
- Compiler optimizations(SW)
- HW design space
- Profiling information for HW/SW tradeoffs

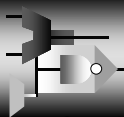
26/10/2001      Reconfigurable Systems      31



## Retargetable System

- Yes!!
- Platform described in ADL by
  - Type of processor
  - Characteristics of the FPGA
  - Memory

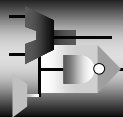
26/10/2001      Reconfigurable Systems      32



## Why loops

- Significant portion of the execution time
- Compact implementation of loops

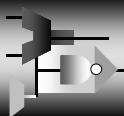
26/10/2001      Reconfigurable Systems      33



## Loop reprocessing

- Profile target architecture
- Extract loops
- Synthesize HW of loops
- Multiple HW structures
  - Loop unrolling
  - Procedure inlining
  - Branch trimming


26/10/2001      Reconfigurable Systems      34



## Global cost function

- Maximize overall performance
- What to include?
  - SW execution times
  - HW execution times
  - Entry times for HW implementations
  - Exit times for HW implementations
  - Configuration times

26/10/2001      Reconfigurable Systems      35



## Algorithm loops

- Loop Entry Profiling(LED)
- Interesting Loop Detection(ILP)
- Intra Loop Selection
- Inter Loop Selection

26/10/2001      Reconfigurable Systems      36

## Loop Profiling

- Identify all loops
- Trace all the loop entries
- Compression

26/10/2001 Reconfigurable Systems 37

## Interesting Loop Selection

- Percentage contribution to application time

The bar chart displays the percentage contribution of different loop categories to the total application time. The categories include: loops, total time, and various nested loop structures. The legend indicates different levels of nesting and branching, such as 'a: dot image', 'com: resion', 'enc: ing', 'ee: ing', 'ene: er', and 'e: ac: ertion'.

26/10/2001 Reconfigurable Systems 38

## Intra Loop Selection

- Choose the best HW solution that fits the FPGA. Execution times.
- Place infreq. branches in SW for too big HW solutions
- Keep the pure SW solution
- Leave configuration times out

26/10/2001 Reconfigurable Systems 39

## Stop

- Why do we leave out configuration time?

The number of configurations for the loops are not available yet. We haven't made the HW/SW partitioning yet.

26/10/2001 Reconfigurable Systems 40


## Inter Loop Selection

- One SW and one HW solution
- Interactions between loops
- Too big search area ( $2^n$ )
- Group loops into clusters
  - Competing loops
  - Nested loops, branches
- Exhaustive search within the clusters

26/10/2001 Reconfigurable Systems 41

## Algorithm

26/10/2001 Reconfigurable Systems 42




## □ **uestions**

? What is the difference between spatial and temporal partitioning?

? How can we speed up execution even more?

26/10/2001 Reconfigurable Systems 43



## □ **onclusions**

- Partitioning in reconfigurable logic(FPGA)
- Spatial and Temporal partitioning
- Config-Exe-Recon-exe .....
- Execute and reconfigure in parallel
- Speedup embedded CPU with HW in FPGA

26/10/2001 Reconfigurable Systems 44