TECHNISCHE UNIVERSITÄT WIEN

Institut für Computertechnik
Institute of Computer Technology

Dynamic Partial Reconfiguration in FPGAs

A Tutorial

- By Andreas Dejmek

Supervised by Nima TaheriNejad

Outline

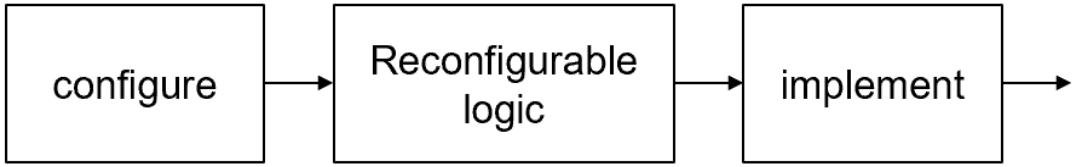
- Basic principles
- Types of configuration interfaces
- Design flow
- Application

Outline

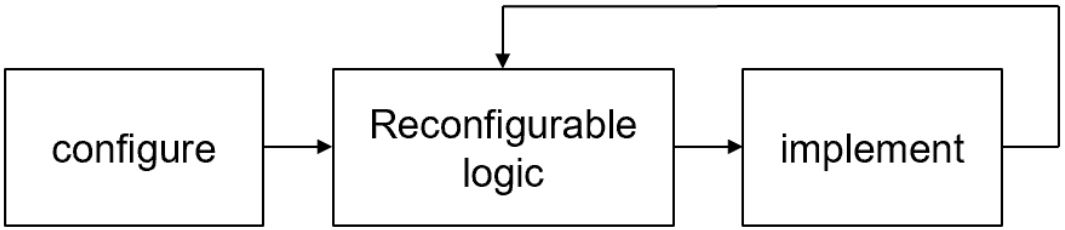
- Basic principles
- Types of configuration interfaces
- Design flow
- Application

Partial reconfiguration

- Static: FPGA is stopped during reconfiguration



- Dynamic: reconfiguration during run-time
  - Only parts of a system might be updated



Source: [1]

Motivation

- Why is DPR useful?
- What can it be used for?

- Efficient resource utilization
- Save reconfiguration time
- Save memory

- In-field updates
- Fault-redundant systems

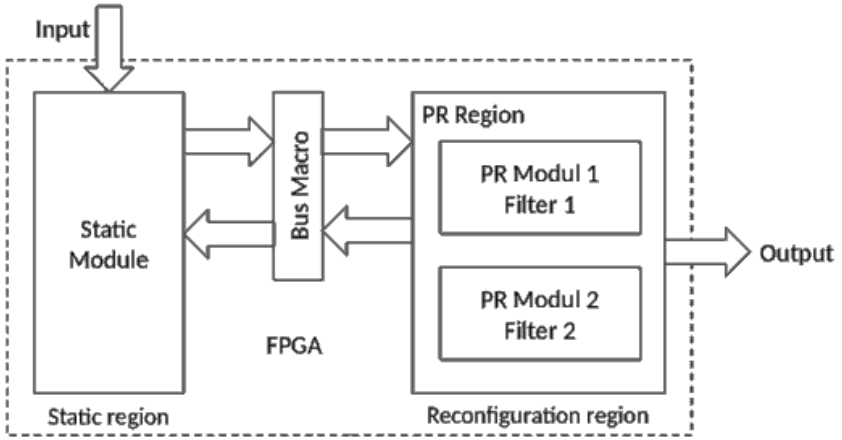
Basic methods

Difference-based

- Allow small design changes
  - LUTs equations
  - I/O standards
  - RAM content
- Bitstreams only contain differences

Module-based

- FPGA area is split into regions:
  - Static region
  - Partial reconfiguration regions



Source: [1]

Outline

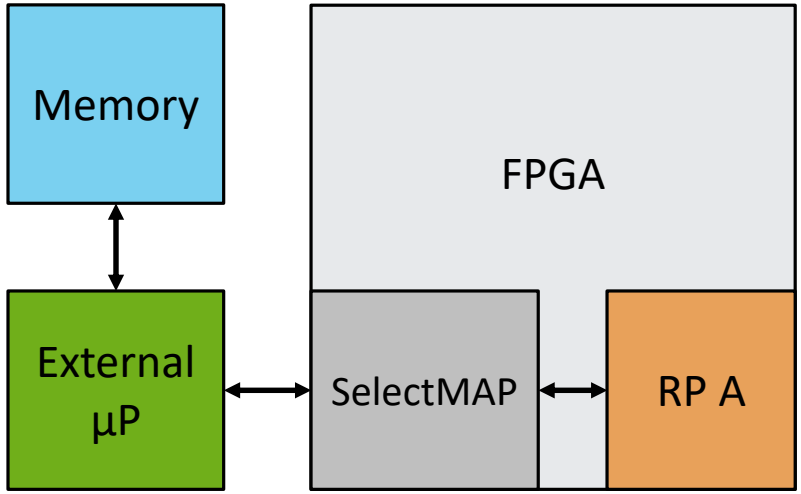
- Basic principles
- Types of configuration interfaces
- Design flow
- Application

Types of configuration interfaces

- Joint Test Action Group *(JTAG)*
- Serial mode
- SelectMAP
- Internal configuration access port *(ICAP)*
- Processor configuration access port *(PCAP)*
- Media configuration access port *(MCAP)*

SelectMAP

- Slave/master modes



- Bus width: 8-bit, 16-bit or 32-bit bidirectional
  - Auto detection
  - Defined during design phase

- set_property CONFIG_MODE <value> [current_design]

Source: [10]

SelectMAP – reconfiguration

- BitGen *persist* option must be set
  - Otherwise, data pins → user I/Os
- set_property BITSTREAM.CONFIG.PERSIST <value> [current_design]

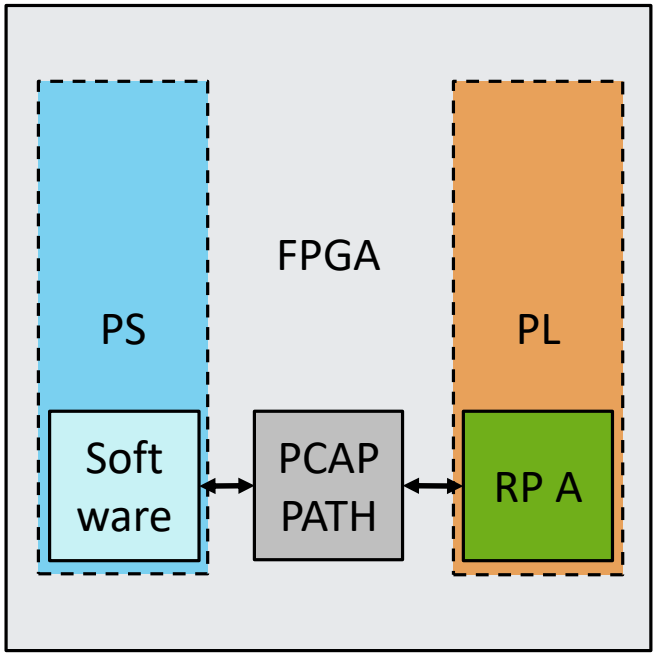
Source: [4]

ICAP

- Internal version of SelectMAP
  - Except ICAP has dedicated data I/Os
- Is a primitive
  - ICAPE2 for Xilinx® 7 Series / Zynq-7000 SoC
  - ICAPE3 for UltraScale / UltraScale+
- Initial configuration not possible
  - Instantiation of primitive

PCAP

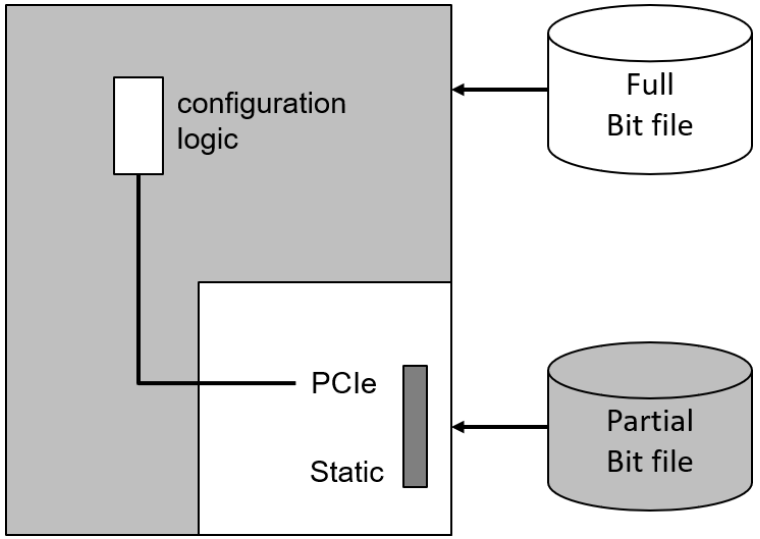
- Configure PL from PS
- Uses DMA controller
- Xilinx® provides device drivers
  - Linux OS
- Initial configuration possible
  - Boots from PS



Source: [14]

MCAP

- Configuration through PCIe block
- Tandem PCIe configuration
- Reconfiguration
  - PCIe link remains active
- Initial configuration not possible



Source: [4]

Supported configuration interfaces

Configuration Mode	7 Series	Zynq-7000	UltraScale	UltraScale+	Zynq UltraScale+ MPSoC
external:					
JTAG	Yes	Yes	Yes	Yes	Yes
Serial mode	Yes	N/A	Yes	Yes	N/A
SelectMAP	Yes	N/A	Yes	Yes	N/A
internal:					
ICAP	Yes	Yes	Yes	Yes	Yes
PCAP	N/A	Yes	N/A	N/A	Yes
MCAP	N/A	N/A	Yes	Yes	Yes

Source: [4]

Performance throughput

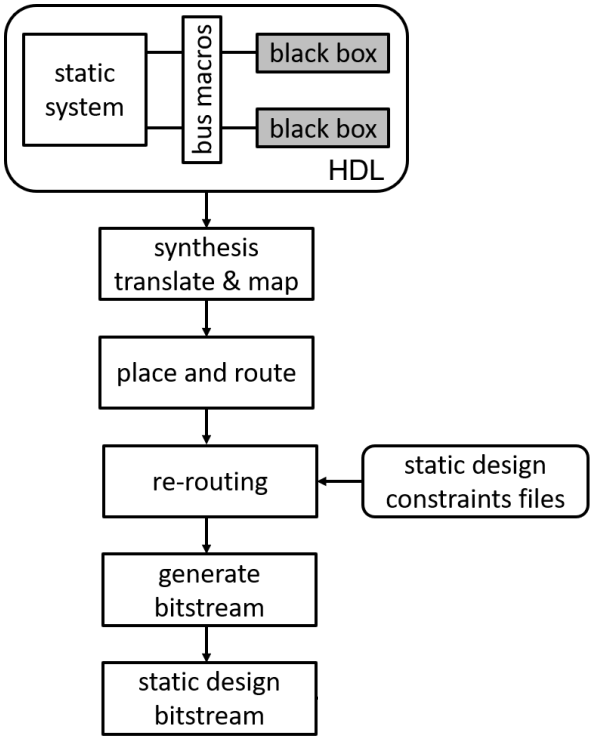
configuration mode	max. clock rate	data width	max. bandwidth
Serial mode	100 MHz	1-bit	12.5 MB/s
JTAG	66 MHz	1-bit	8.25 MB/s
SelectMAP **ICAP** **PCAP**	100 MHz	32-bit	400 MB/s
MCAP	200 MHz	32-bit	800 MB/s

Source: [4, 16]

Outline

- Basic principles
- Types of configuration interfaces
- Design flow
- Application

Design flow



Source: [17]

Design restrictions

- Restrictions for Pblocks

Affects architectures before UltraScale:

- No clock or clock modifying logic
  - PLLs
  - Clock buffers
- I/Os and I/O related components

Outline

- Basic principles
- Types of configuration interfaces
- Design flow
- Application

Application – Image processing

- Using dynamic partial reconfiguration to:

- Implement two morphological functions
  - Erosion
  - Dilation

Source: [19]

Application – Image processing

Original

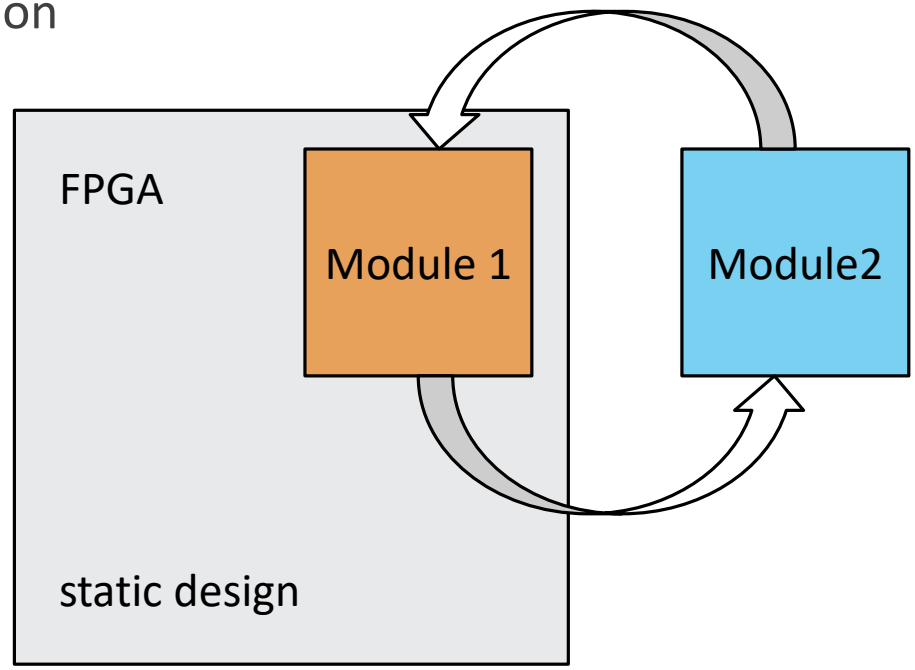
Eroded

Dilated

Source: [19]

Application – Image processing

- Using ICAP for reconfiguration
- Module
  - Erosion
  - Dilation



Source: [19]

Application – Conclusion

bitstream	size (bytes)	configuration time (µs)
full	250.680	627,2
partial	35.840	92,3
Savings	**85,7%**	**85,3%**

$$t[\mu s] = \frac{bitstream\_size * 8}{32 * 100}$$

$$\begin{cases} 32\ bit \ldots bus\ width\ of\ ICAP \\ 100\ MHz \ldots clock\ frequency\ of\ ICAP \end{cases}$$

Source: [19]

Sources

- [1] W. Lie and W. Feng-yan, “Dynamic partial reconfiguration in fpgas,” in 2009 Third International Symposium on Intelligent Information Technology Application, vol. 2. IEEE, Nov 2009, pp. 445–448.
- [2] K. Vipin and S. A. Fahmy, “Fpga dynamic and partial reconfiguration: A survey of architectures, methods, and applications,” ACM Comput. Surv., vol. 51, no. 4, pp. 72:1–72:39, Jul. 2018.
- [3] E. Eto, “Difference-based partial reconfiguration,” XAPP290 (v2.0), Dec. 2007.
- [4] Xilinx, “Vivado design suite user guide – partial reconfiguration,” UG909 (v2019.1), Jun. 2019.

Sources

- [5] D. Koch, J. Torresen, C. Beckhoff, D. Ziener, C. Dennl, V. Breuer, J. Teich, M. Feilen, and W. Stechele, “Partial reconfiguration on fpgas in practice — tools and applications,” in ARCS 2012. IEEE, 2012, pp. 1–12.
- [6] D. Koch, Introduction. New York, NY: Springer New York, 2013, pp. 1–42.
- [7] Xilinx, “Virtex-5 fpga - configuration user guide,” UG191 (v3.12), May 2017.
- [8] Xilinx, “Vivado design suite - user guide: Programming and debugging,” UG908 (v2012.2), Jul. 2012.
- [9] Xilinx, “7 series fpgas configuration - user guide,” UG470 (v1.13.1), Aug. 2018.
- [10] Xilinx, “Vivado design suite - properties reference guide,” UG912 (v2016.1), Apr. 2016.

Sources

- [11] Xilinx, “Vivado design suite 7 series fpga and zynq-7000 soc - libraries guide,” UG953 (v2019.2), Oct. 2019.
- [12] Xilinx, “Ultrascale architecture configuration - user guide,” UG570 (v1.11), Sep. 2019.
- [13] Xilinx, “Partial reconfiguration controller v1.3 – logicore ip product guide,” PG193, Apr. 2018.
- [14] Xilinx, “Zynq-7000 soc - technical reference manual,” UG585 (v1.12.2), Jul. 2018.
- [15] Xilinx, “Zynq ultrascale+ device - technical reference manual,” UG1085 (v2.1), Aug. 2019.
- [16] Xilinx, “Partial reconfiguration of a hardware accelerator on zynq-7000 all programmable soc devices,” XAPP1159 (v1.0), Jan. 2013.

Sources

- [17] P. Sedcole, B. Blodget, T. Becker, J. Anderson, and P. Lysaght, “Modulardynamic reconfiguration in virtex fpgas,”IEE Proceedings - Computersand Digital Techniques, vol. 153, no. 3, pp. 157–164, May 2006.
- [18] A. Hassan, R. Ahmed, H. Mostafa, H. A. H. Fahmy, and A. Hussien, “Performance evaluation of dynamic partial reconfiguration techniques for software defined radio implementation on fpga,” in 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Dec 2015, pp. 183–186.
- [19] R. Hentati, M. Hentati, Y. Aoudni, and M. Abid, “The implementation ofbasic morphological operations on fpga using partial reconfiguration,” inInternational Image Processing, Applications and Systems Conference, Nov 2014, pp. 1–5.

Questions?

THANK YOU FOR YOUR ATTENTION!