
University of
L'Aquila - Italy
IWES 2020
5th Italian Workshop on Embedded Systems
DEWS
CENTER OF EXCELLENCE
disim
DPR time estimation:
a static timing approach
G. Valente, G. D'Andrea, T. Di Mascio, L. Pomante
DISIM, DEWS
Università degli Studi dell'Aquila
L'Aquila, Italy

University of L'Aquila - Italy

Outline

DEWS CENTER OF EXCELLENCE disim

01 The Dynamic Partial Reconfiguration process

02 SoA study and analysis

03 The proposed approach

The SoA study → Reference platform → The SoA analysis → The proposed approach → Validation

University of L'Aquila - Italy

Outline

DEWS CENTER OF EXCELLENCE disim

01 The Dynamic Partial Reconfiguration process

02 SoA study and analysis

03 The proposed approach

The SoA study → Reference platform → The SoA analysis → The proposed approach → Validation

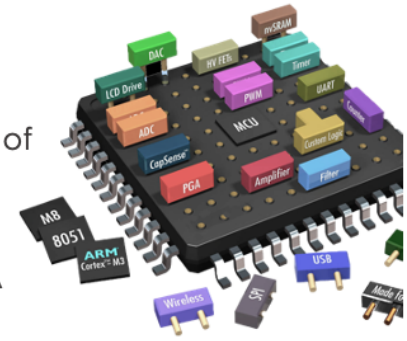
University of L'Aquila - Italy

The Dynamic Partial Reconfiguration process

DEWS CENTER OF EXCELLENCE disim

The DPR is an FPGA process that:

- ✓ Allows to **reconfigure, at run-time,** a portion of HW
- ✓ Can be considered established only when a reconfiguration file is loaded within the FPGA reconfiguration area;
- ✓ Involves more than one platform component;



The DPR process requires a certain amount of time to transfer the reconfiguration file, commonly called bitstream (BS), from the external memory to the FPGA reconfiguration area (i.e., FPGA reconfiguration memory).

University of L'Aquila - Italy

The Dynamic Partial Reconfiguration process

DEWS CENTER OF EXCELLENCE disim

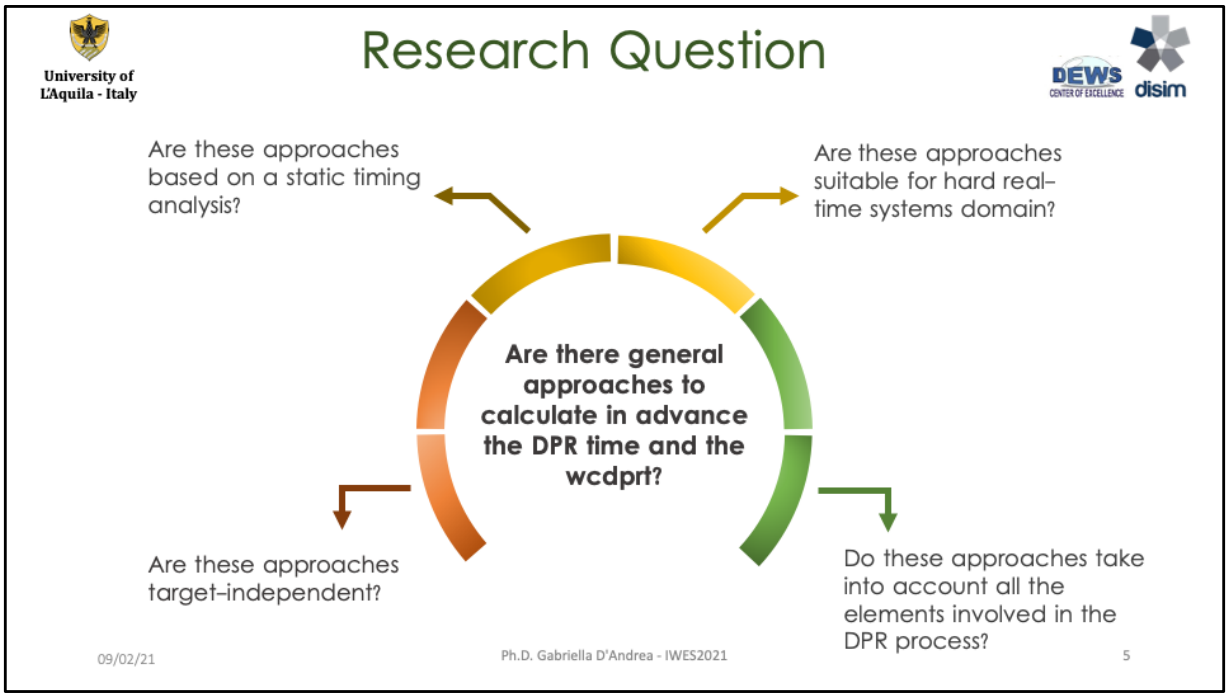
- In the hard real-time systems domain, where it is mandatory to guarantee the worst-case response time (wcrt) of each process, the usage of DPR introduces **advantages** and **disadvantages:**

DISADVANTAGES

There can be the non-negligible reconfiguration time impact and a **lack of a general approach to calculating its worst-case bound**.

The DPR can be used to improve task performances in terms of timing and/or power.

ADVANTAGES



At this time our question is: are there general approaches to calculate the DPR time before the start of the reconfiguration process?
And, if such approaches exist, are they suitable for hard real-time systems? Are these approaches based on a static timing analysis (i.e., an analysis that can be performed at design time)? Are these approaches target-independent, hence potentially applicable to all DPR compliant platforms? And finally, do these approaches take into account all the elements involved in the DPR process?

University of L'Aquila - Italy

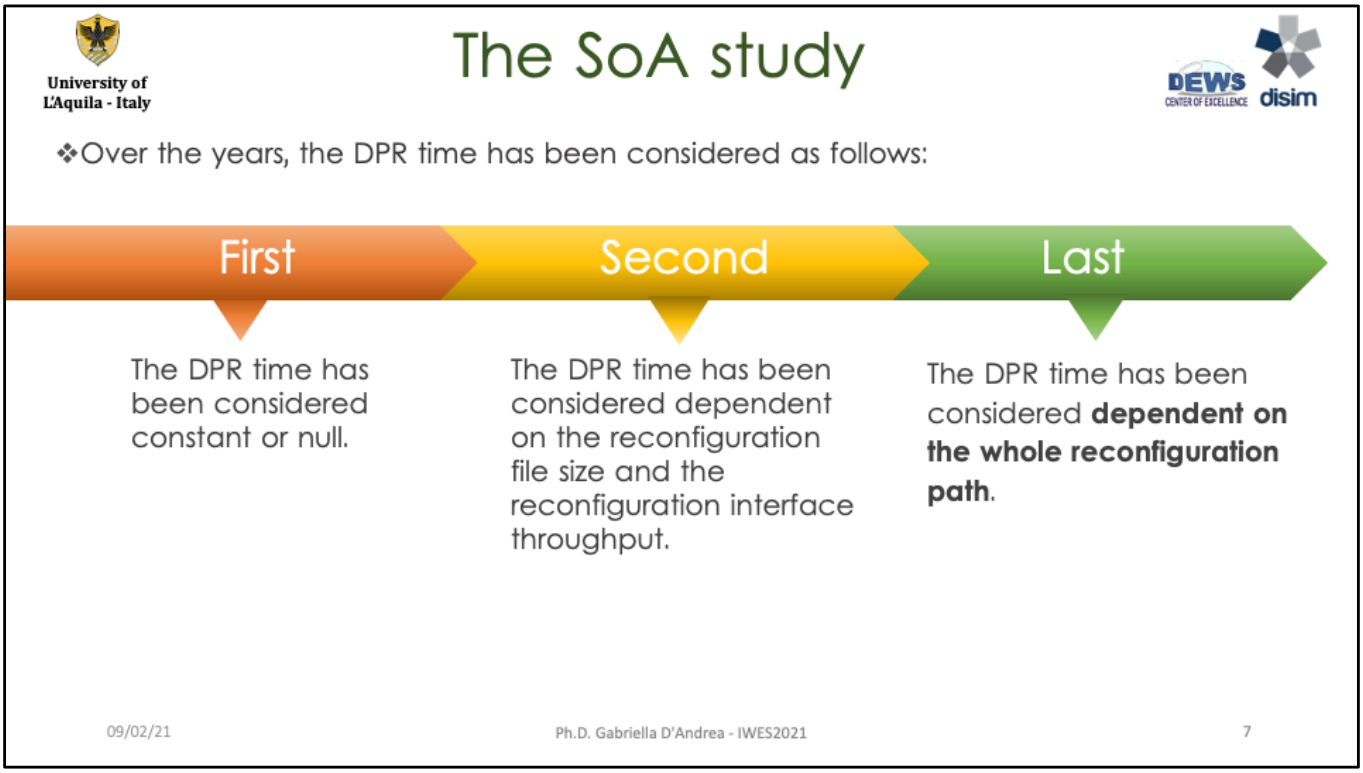
Outline

DEWS CENTER OF EXCELLENCE disim

01 The Dynamic Partial Reconfiguration process

02 SoA study and analysis

03 The proposed approach



Our DPR time SoA study highlights that does not exist an approach that, considering the whole reconfiguration path (path used to transfer the BS from the external memory to the FPGA reconfiguration memory), allows to calculate in advance the DPR time and it is potentially applicable to all DPR compliant platforms.
Hence at this point, we have started working on the reconfiguration path.

University of L'Aquila - Italy

Outline

DEWS CENTER OF EXCELLENCE disim

01 The Dynamic Partial Reconfiguration process

02 SoA study and analysis

03 The proposed approach

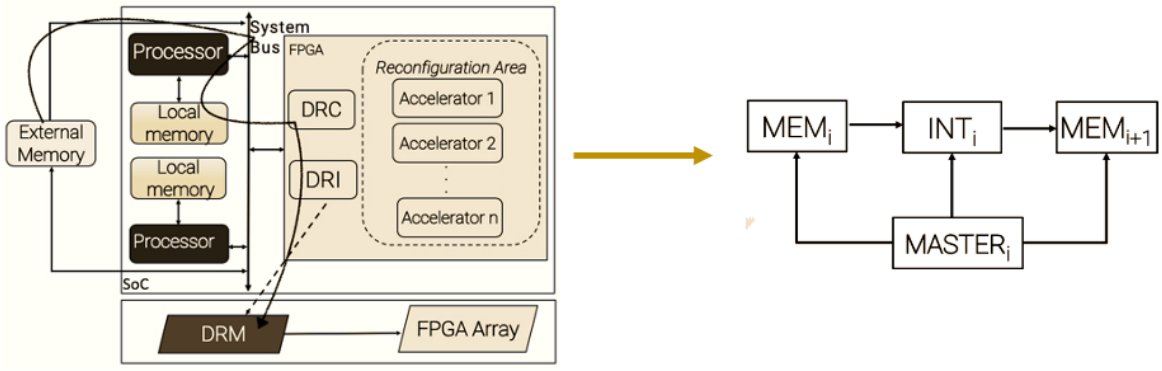
The SoA study → Reference platform → The SoA analysis → The proposed approach → Validation

University of L'Aquila - Italy

Method to analyse the SoA

DEWS CENTER OF EXCELLENCE disim

- Development of general reference platform with the identification of a **reconfiguration path** (RP) divided in **elementary paths** (EPs).



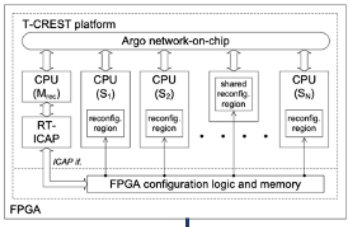
09/02/21 Ph.D. Gabriella D'Andrea - IWES2021 9

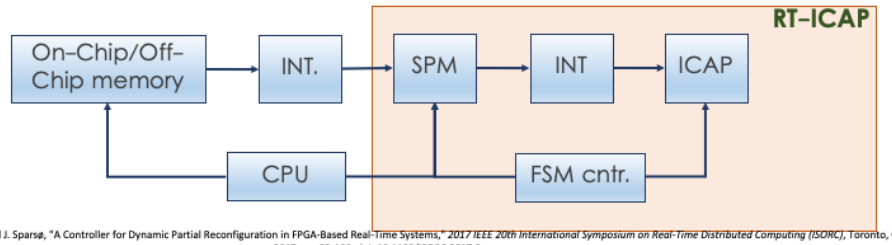
We have developed a general reference platform for DPR identifying the reconfiguration path (indicated in the figure with the black arrow) and dividing it into the elementary path (depicted on the right side of the figure).

University of L'Aquila - Italy

The SoA analysis

A Controller for Dynamic Partial Reconfiguration in FPGA-Based Real-Time System*





*L. Pezzarossa, M. Schoeberl and J. Sparsø, "A Controller for Dynamic Partial Reconfiguration in FPGA-Based Real-Time Systems," *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)*, Toronto, ON, 2017, pp. 92-100, doi: 10.1109/ISORC.2017.3.

09/02/21 Ph.D. Gabriella D'Andrea - IWES2021 10

By instantiating the general DPR compliant platform into the works presented in the DPR time SoA we have studied their method to calculate the reconfiguration time.

University of L'Aquila - Italy

Results of SoA Analysis

DEWS CENTER OF EXCELLENCE disim

- The SoA analysis highlights the **lack** of an approach to bound the worst-case DPR time that, at the same time:
  - is target independent;
  - considers all the RP elements;
  - provides a worst-case DPR time bound

Hardware resources

Approach	Target indep.	All RP elements	WCDPRT	FF	LUT	BRAM
CoRQ	no	yes	yes	552	631	97
RT-ICAP	no	yes	yes	101	245	Encoded Bitstream size
ZyCAP	no	no	no	806	620	0

09/02/21 Ph.D. Gabriella D'Andrea - IWES2021 11

Some of the works that we have studied are reported in the table Hardware Resources.
In general most of the studied works require the usage of custom DPR controllers that make them target-dependent increasing both the area occupied in the FPGA and the power consumed.

University of L'Aquila - Italy

Outline

DEWS CENTER OF EXCELLENCE disim

01 The Dynamic Partial Reconfiguration process

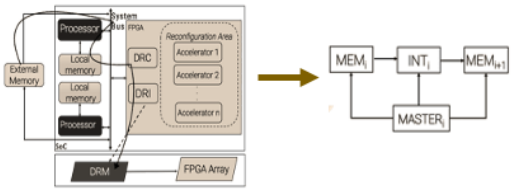
02 SoA study and analysis

03 The proposed approach

University of L'Aquila - Italy

The proposed approach

DEWS CENTER OF EXCELLENCE disim



$$t_{DPR} = f_M\,(BS_{size}, TP_{i,j}, RP_{M_{i,j}})$$
$$i = 1, \ldots . N \qquad j = 1, \ldots F_i$$

- The BSs can be transferred in chunk of different size ($size_{chunk_{i,j}}$) along with the EPs

Hypothesis:

Hp$_1$) current transfers (i.e., pipelined) among EPs are not allowed; → f_M can be written as a sum

Hp$_2$) $size_{chunk_{ij}}$ is constant within each EP → $F_i = \dfrac{BS_{size}}{size_{chunk_i}}$

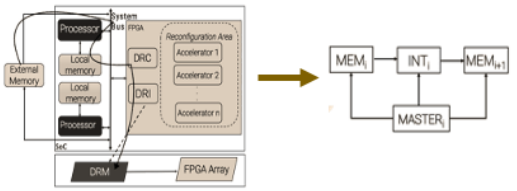
09/02/21 Ph.D. Gabriella D'Andrea - IWES2021 13

Considering the general DPR compliant platform and the elementary path the DPR time (t_{DPR}) can be calculated with the formula on the left side of the slides; where N is the number of elementary paths (EPs) that compose the reconfiguration path and F_i is the number of chunks to be transferred within the i-th EP. In the formula, t_{DPR} is a function (f_M) of : (i) the BS_{size},(ii) the $TP_{i,j}$, that represents,∀ EP_i, the throughput associated to the transfer of each $chunk_{i,j}$ within EP_i, and (iii) $RP_{Mi,j}$, that represents the RP policy adopted to manage the transfer of chunks among adjacent EPs.

University of L'Aquila - Italy

The proposed approach

DEWS CENTER OF EXCELLENCE disim



$$t_{DPR} = \sum_{i=1}^{N}\left[\sum_{j=1}^{F_i}\left(\frac{size_{chunk_{i,j}}}{TP_{i,j}}\right)\right] = \sum_{i=1}^{N}\left[\left(\frac{BS_{size}}{size_{chunk_i}} \cdot \frac{size_{chunk_i}}{TP_i}\right)\right] = \sum_{i=1}^{N}\frac{BS_{size}}{size_{chunk_i}} \cdot t_{M_i}$$

$$t_{M_i} = f_t\left(t_{MEM_i}, t_{MEM_{i+1}}, t_{INT_i}, t_{MST_i}\right)$$

$$i = 1, \ldots . N \qquad j = 1, \ldots F_i$$

09/02/21 Ph.D. Gabriella D'Andrea - IWES2021 14

Since at design-time we know both the BS size and the chunk size we can calculate the reconfiguration time considering the features of each component of each elementary path. For example, we can consider the size of the memories, or the interconnections frequencies, or the master command queue depth.

University of L'Aquila - Italy

Outline

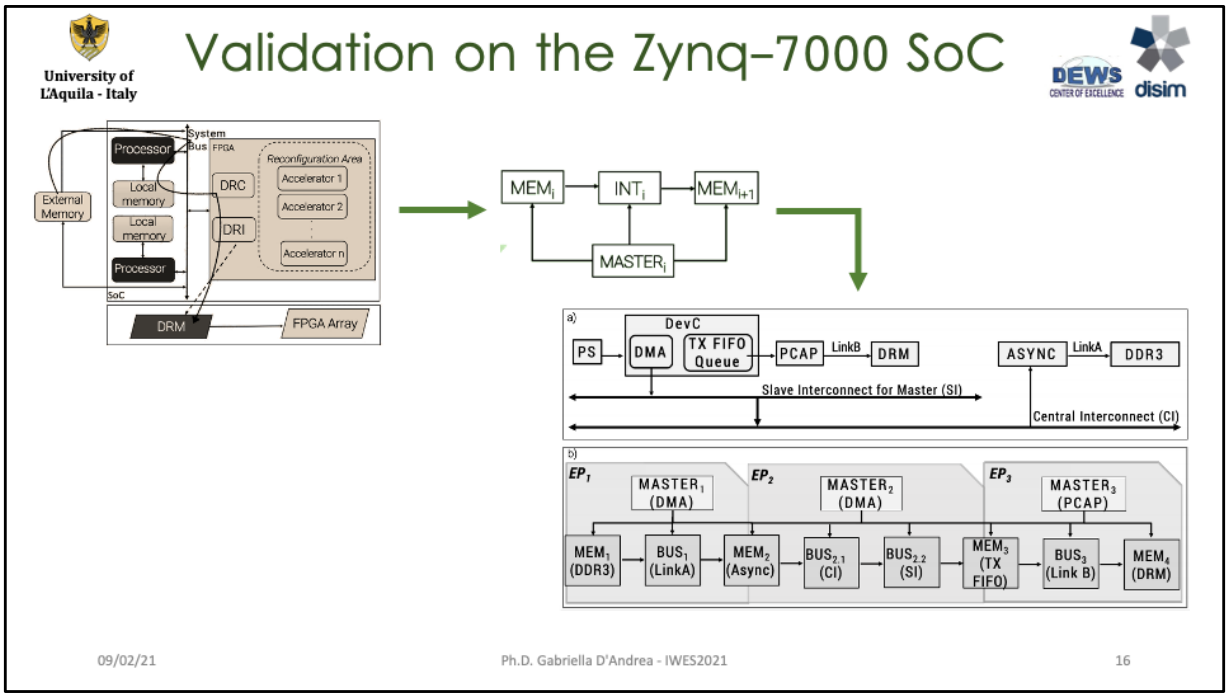
DEWS CENTER OF EXCELLENCE disim

01 The Dynamic Partial Reconfiguration process

02 SoA study and analysis

03 The proposed approach

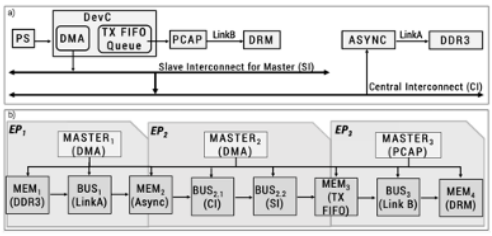
The SoA study → Reference platform → The SoA analysis → The proposed approach → Validation



Using the documentations of the Zynq-7000 SoC we have found three elementary paths among which the BS is transferred.

University of L'Aquila - Italy

Validation on the Zynq-7000 SoC



$$t_{DPR}^{CAL} = \sum_{i=1}^{N} \frac{BS_{size}}{size_{chunk_i}} \cdot t_{M_i} = \sum_{i=1}^{3} \left(\frac{BS_{size}}{size_{chunk_i}} \cdot t_{M_i} \right)$$

$$\approx BS_{size} \left(\frac{1}{freq_2 \cdot beat_{size_2}} + \frac{1}{BW_{PCAP}} \right)$$

Results for Bs_{size} = 857740 Bytes and the accuracy(acc%) obtained using the following formula:

$$acc = \frac{|\, t_{DPR}^{CAL} - t_{DPR}^{ACT} \,|}{t_{DPR}^{ACT}}$$

$beat_{size_2}$ (bytes)	1 (worst case)	2	3	4
$t_{DPR}^{CAL} - t_{DPR}^{ACT}$ (ms)	2	-1.2	-2.3	-2.9
acc%	30.3	18.1	34.8	43.9

G. Valente, T. Di Mascio, G. D'Andrea and L. Pomante, "Dynamic Partial Reconfiguration Profitability for Real-Time Systems," in IEEE Embedded Systems Letters, doi: 10.1109/LES.2020.3004302.

09/02/21 Ph.D. Gabriella D'Andrea - IWES2021 17

In this work, considering having exclusive access to the reconfiguration path (i.e., one actor that accesses the memories and the shared buses) we have calculated the DPR time. More details about that approach and the DPR time calculation can be found in our work titled Dynamic Partial Reconfiguration Profitability for Real-Time Systems (G. Valente, T. Di Mascio, G. D'Andrea and L. Pomante, "Dynamic Partial Reconfiguration Profitability for Real-Time Systems," in IEEE Embedded Systems Letters, doi: 10.1109/LES.2020.3004302).

University of L'Aquila - Italy

Conclusion

DEWS CENTER OF EXCELLENCE disim

- The proposed approach:
  - is potentially applicable to all the possible reconfigurable paths in any of the available DPR–compliant platforms
  - allows calculating a WCDPRT suitable in the hard real–time systems domain.

Hardware resources

Approach	Target indep.	All RP elements	WCDPRT	FF	LUT	BRAM
CoRQ	no	yes	yes	552	631	97
RT–ICAP	no	yes	yes	101	245	Encoded Bitstream size
ZyCAP	no	no	no	806	620	0
The proposed approach	**Yes**	**Yes**	**Yes**	**0**	**0**	**0**

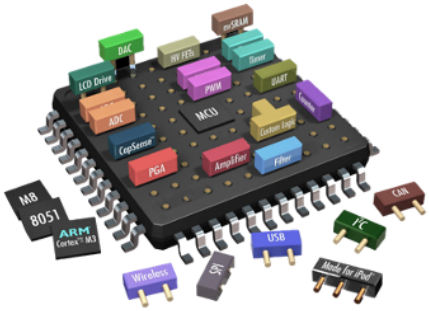
University of L'Aquila - Italy

Future works

DEWS CENTER OF EXCELLENCE disim

- To validate the proposed approach through more complex scenarios, which include access to shared elements.
- To validate the proposed approach by means of different reconfigurable platforms, such as those of Intel.
- To develop a runtime manager able to make use of the proposed approach to predict the DPR time, with the goal to refine task scheduling with hardware accelerated tasks.
- To investigate the DPR controllers features to allow maximum use of their capabilities during the reconfiguration process management.

University of L'Aquila - Italy



DEWS CENTER OF EXCELLENCE disim

Thank you for your attention!

Any questions?

Contact:
gabriella.dandrea@graduate.univaq.it