Image and Video Processing and Communications Laboratory

Dynamically Reconfigurable Management of Energy, Performance, and Accuracy applied to Digital Signal, Image, and Video Processing Applications

Daniel Llamocca

Electrical and Computer Engineering Department

The University of New Mexico

December 2nd, 2011

Outline

- Motivation
- Related Work
- Thesis Statement
- Contributions
- General Approach
- General implementation details
- Digital signal, image, and video processing applications:
  - General Implementation details
  - Pixel Processor
  - 1D FIR Filter
  - 2D FIR Filter/Filterbank
- Conclusions

Motivation

Digital signal, image, and video processing systems can be characterized by three properties:

Energy, Performance, and Accuracy (EPA).

*The controlling of these variables at run-time is defined as **Dynamic Energy-Performance-Accuracy (EPA) management**.*

Dynamic EPA management will enable us to deliver:

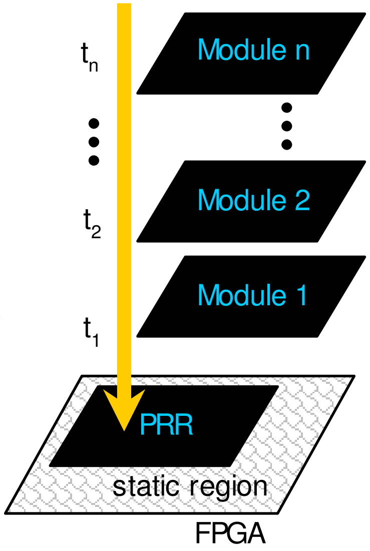
- *A dynamically self-adaptive system (by dynamic allocation of computational resources and dynamic frequency control) that satisfies time-varying EPA requirements.*
- *An Optimal resulting realization: We want to investigate optimal solutions that can meet dynamic EPA requirements . The system should **minimize energy consumption, and at the same time maximize performance and accuracy**, while satisfying the given EPA requirements.*

Motivation

Dynamic Energy-Performance-Accuracy management can rely on Dynamic Partial Reconfiguration (DPR) and Dynamic Frequency Control on FPGAs.

Dynamic Partial Reconfiguration

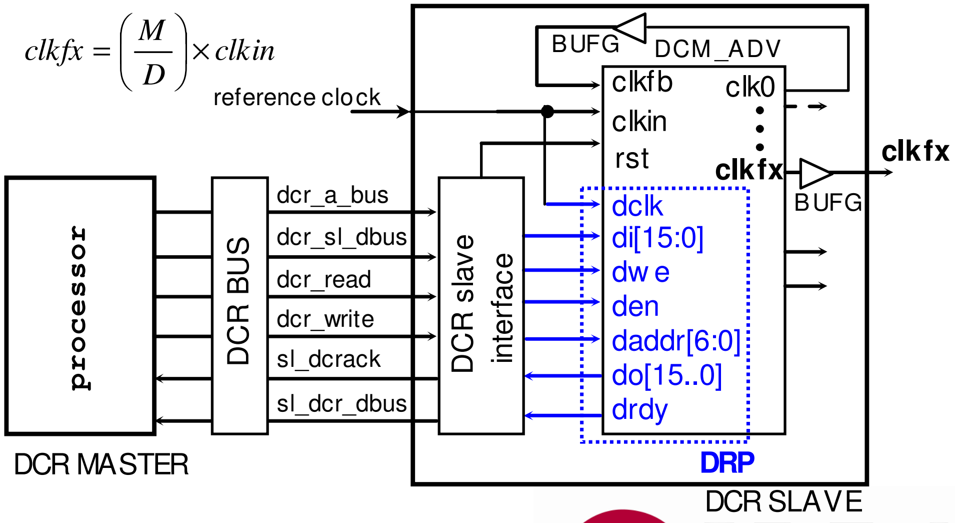
DPR technology enables the adaptation of hardware resources by modifying or switching off portions of the FPGA while the rest remains intact, continuing its operation. To perform DPR, the Partial Reconfiguration Region (PRR) must be defined. The PRR is dynamically reconfigured via the internal configuration access port (ICAP).



Dynamic Frequency Control

Digital Clock Managers (DCMs) inside FPGAs provide a wide range of clock management features.

The Dynamic Reconfiguration Port (DRP) of the DCM enables dynamic control of the frequency and phase. We can use it to dynamically adjust the frequency without reloading a new bitstream to the FPGA.



$$clkfx = \left(\frac{M}{D}\right) \times clkin$$

Motivation

The system can then carry out independent tasks in time:

task 1, task 2, ….

Examples:

- *Task 1: A video processing system is asked to deliver real time performance at 30 frames per second (fps) on limited battery life that will also need to operate for at least 10 hours. This is a multi-objective optimization problem. If solutions are found, pick the system realization with the highest precision.*

- *Task 2: Now, suppose that we are asked to deliver performance at 100 fps at some minimum level of accuracy (60dB). In this case, we can select the hardware realization with the lowest energy requirements while meeting the performance and accuracy constraints.*

Related work (1 of 3)

Image processing with DPR:

- DPR implementation of mean and median filters [Bhandari09], [Raikovich10].
- Fingerprint image processing algorithms whose stages (segmentation, normalization, smoothing, etc.) are multiplexed in time via DPR [Fons10].
- 3D Haar Wavelet Transform DPR implementation by dynamically reconfiguring a 1D HWT thrice [Afandi09].
- JPEG2000 decoder where the blocks are dynamically swapped [Bouchoux04]
- All these works are DPR implementations that exhibit some resemblance to our work. However they did not explore the EPA space.

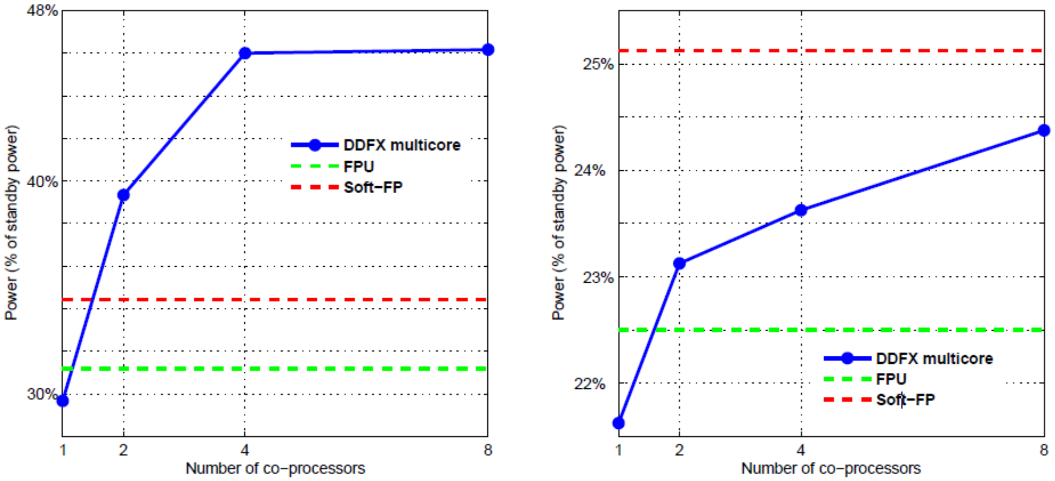
CHREC (NSF Center for High-Performance Reconfigurable Computing):

- Acceleration of the Partial Reconfiguration Process (e.g. bitstream rellocator, high level PR description for fast PR implementation, platform for rapid deployment of PR embedded systems, using hard macros to reduce FPGA compilation time).
- Adaptive filtering, optical flow static implementations (no DPR)
- JTAG encoder/decoder (modules are swapped via DPR)
- No exploration of the EPA space via DPR.

Related work (2 of 3)

DPR Application in Dynamic Arithmetic [Vera08]:

- The use of DPR provided a low-energy example where the use of dynamic dual fixed-point (DDFX) arithmetic was shown to perform as well as double floating point (FP) in a Linear Algebra example.



(a) Virtex 2 Pro - Microblaze.

(b) Virtex 4 - PowerPC405.

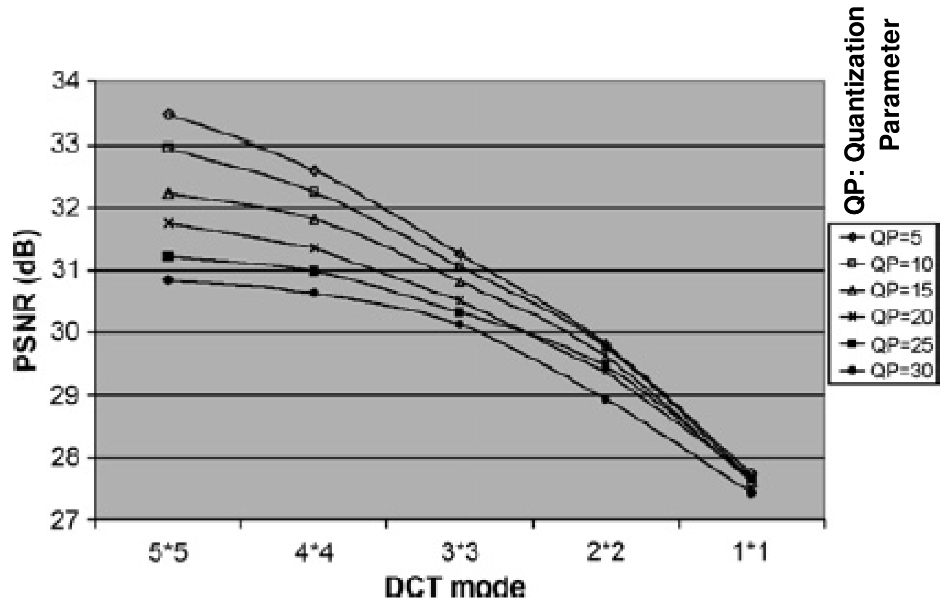
Figure 6.4: Dynamic power consumption.

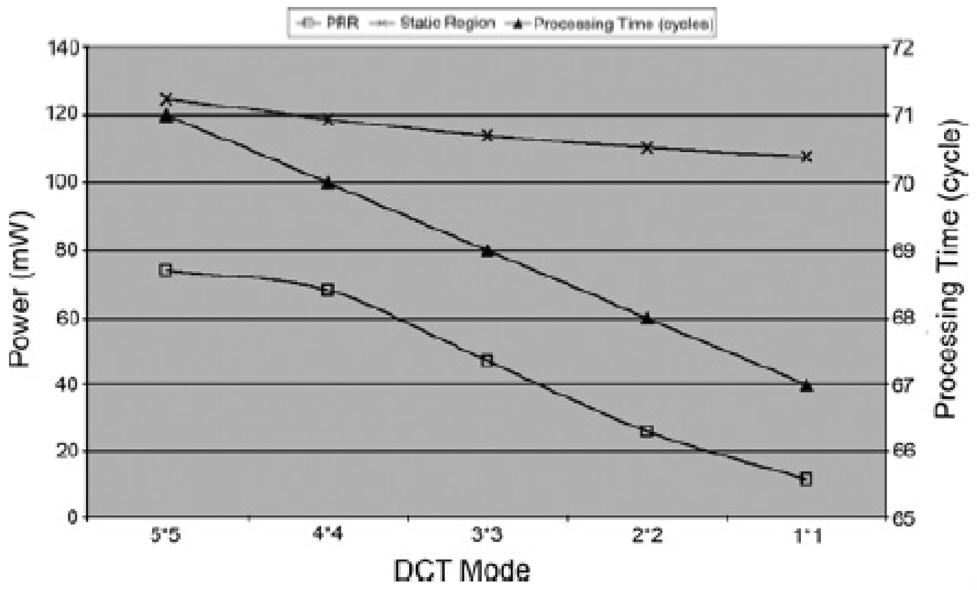
- DDFX maintains a performance advantage with respect to FP when reconfiguring once every 10000 operations or less (i.e., DDFX can change precision 250 times per second or switch operations 150 per second.
- Arithmetic cores were measured in terms of their power, performance, and precision.
- A model was formulated that relates **power, performance, and precision** of the dynamic arithmetic architecture. It explored the use of DPR to dynamically adjust performance, precision, and power consumption.
- No multi-objective optimization of the Power-Performance-Precision space

Related work (3 of 3)

DPR Application for scalable DCT computation [Huang09]:

- Parameterized Discrete Cosine Transform (DCT) systolic modules (no Distributed Arithmetic approach suitable for FPGAs)
- The system dynamically reconfigures among Discrete Cosine Transform modules of different sizes (e.g., 8x8, 5x5,4x4).
- Different DCT configurations were studied in terms of *power, performance, and accuracy*. A configuration manager can adapt DCTs of different sizes based on *power, performance, and accuracy* constraints.
- Exploration of power, performance, and precision dependence on the DCT size. No multi-objective optimization of the EPA space.





Thesis Statement

This Dissertation develops a dynamic Energy-Performance-Accuracy management framework for Digital Signal, Image, and Video processing applications.

This entails:

1. ***Development and parameterization of efficient and dynamically reconfigurable architectures for the following signal, image, and video processing applications****:*

   DPR Pixel Processor

   DPR 1-D FIR Filter

   DPR 2D FIR Filter

   DPR 2D Filterbank

2. ***Development of a Multi-objective Pareto optimization approach to meet global Energy-Performance-Accuracy (EPA) constraints****.*
3. ***Description of the Pareto-optimal realizations extracted from the EPA space****: We are interested in how the architecture parameters generate Pareto-optimal solutions from the EPA space.*
4. ***Dynamic EPA management to meet time-varying global EPA constraints.*** *The system receives stimuli in the form of EPA constraints and reconfigures itself via DPR and/or dynamic frequency control to meet the EPA constraints.*

Contributions

- *Development of fully-parameterized hardware cores for signal, image, and video processing applications. The architectures are implemented with techniques that minimize the amount of computing resources and take advantage of Dynamic Partial Reconfiguration.*

- *Characterization of the optimal (in the multi-objective sense) hardware realizations from the EPA/PPA space for the architectures presented.*

- *A new framework for dynamic energy/power, performance, and accuracy (EPA/PPA) management based on a multi-objective optimization approach that guarantees low energy, high accuracy, and high performance. The framework is applicable to a wide array of signal, image, and video processing architectures.*

- *Development of hardware systems that support dynamic energy/power, performance, and accuracy management that meet real-time EPA/PPA constraints. On hardware, dynamic EPA/PPA management is based on the run-time control of hardware resources and frequency of operation.*

General approach (1/5)

Steps:

1) *Definition of Objective Functions*

2) *Development of efficient cores*

3) *Parameterization of Hardware Cores*

4) *Multi-objective Pareto Optimization in the EPA Space*

5) *Dynamic management based on real-time EPA constraints*

General approach (2/5)

1) ***Definition of objective functions:*** *Energy, performance, and accuracy are considered the objective functions of system parameters. These properties may have a slightly different definition depending on the application.*

   Energy *can be measured as the total energy spent during the system operation, or the energy spent during an operation (e.g., energy per video frame). In some instances, measuring* ***Power*** *is more useful.*

   Performance *can be measured by: Megasamples per second, frames per second, Megabytes per second, etc.*

   Accuracy *can be measured by: numerical representation, or accuracy with respect to an idealized result (e.g., PSNR).*

2) ***Development of efficient cores:*** *The signal, image, and video processing architectures should use techniques that: i) minimize the amount of computational resources (e.g. LUT-based approaches, Distributed Arithmetic), and ii) make intensive use of DPR.*
   The cores must be implemented in Hardware Description Language (HDL), so that they remain portable across FPGA devices and vendors.

General approach (3/5)

3) ***Parameterization of hardware cores:*** *To achieve a fine control of energy, performance, and accuracy, we require realistic parameterization of the hardware cores (e.g., I/O bit-width, number of parallel cores).*

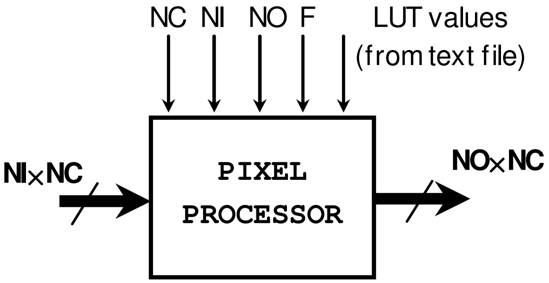
The parameterized HDL code let us create a set of hardware realizations by varying the parameters. Each realization comes with different energy, performance, and accuracy values, which we can control by varying the hardware parameters.

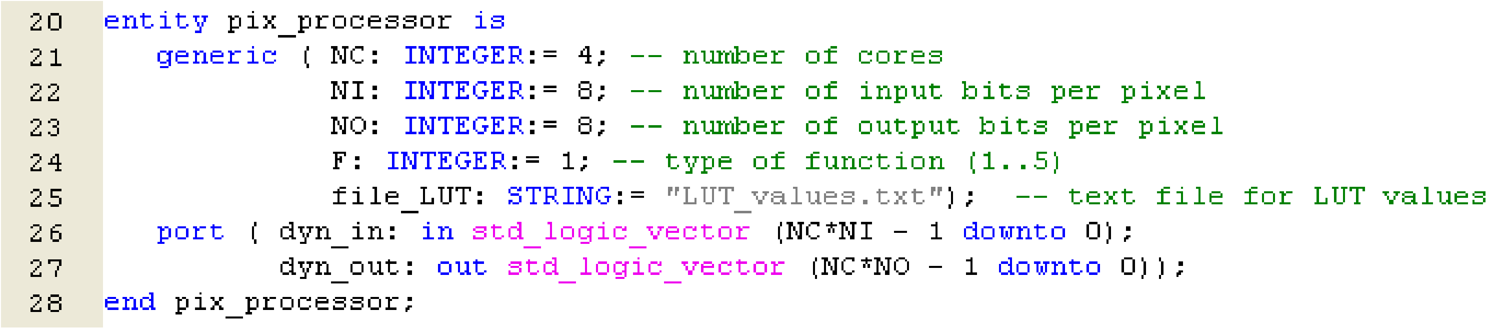
Example: Parameterization of the 'Pixel processor' architecture:

NC (number of cores), NI (number of input bits per pixel),

NO (number of output bits per pixel), F (function to be implemented),

LUT values (text file with LUT values)





```
entity pix_processor is
   generic ( NC: INTEGER:= 4; -- number of cores
             NI: INTEGER:= 8; -- number of input bits per pixel
             NO: INTEGER:= 8; -- number of output bits per pixel
             F: INTEGER:= 1; -- type of function (1..5)
             file_LUT: STRING:= "LUT_values.txt");  -- text file for LUT values
   port ( dyn_in: in std_logic_vector (NC*NI - 1 downto 0);
          dyn_out: out std_logic_vector (NC*NO - 1 downto 0));
end pix_processor;
```

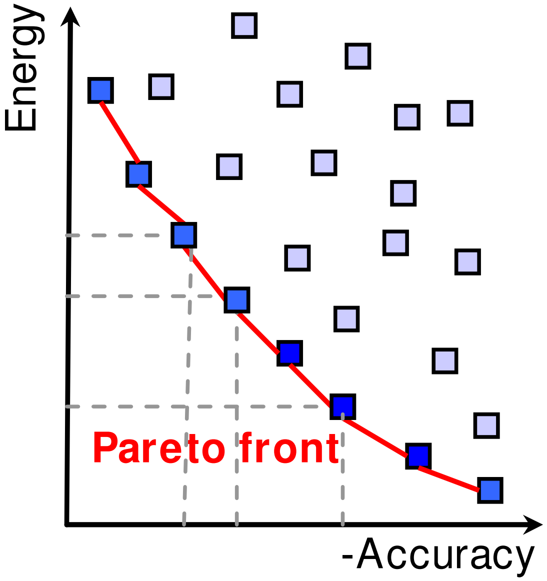
General approach (4/5)

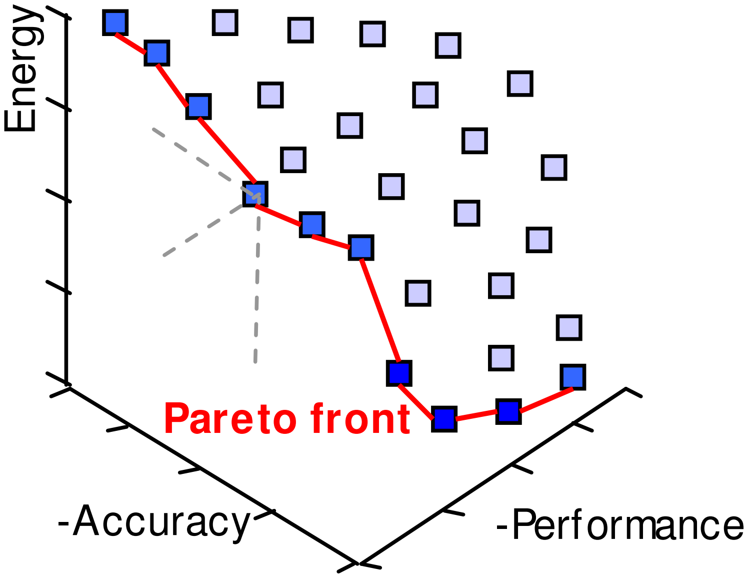
*4) **Multi-objective Pareto Optimization in the EPA Space:** The Energy-Performance-Accuracy (EPA) space is represented by a set of hardware realizations along with their EPA values.*

An optimal hardware realization is defined as the one that minimizes energy, while maximizing performance and accuracy.

We are interested in the set of optimal realizations from the EPA space. We want to find a subset whose EPA values cannot be improved by any other realization for all three (EPA). These realizations are called optimal in the Pareto (multi-objective) sense.

*The Energy-Performance-Accuracy space is shown along with the **Pareto-optimal** points. In some cases, we may want to explore a space of just 2 variables, e.g., the Energy-Accuracy space.*





General approach (5/5)

5) ***Dynamic management based on real-time EPA constraints:*** *Once the Pareto front has been extracted, we can cast optimization problems based on EPA constraints.*

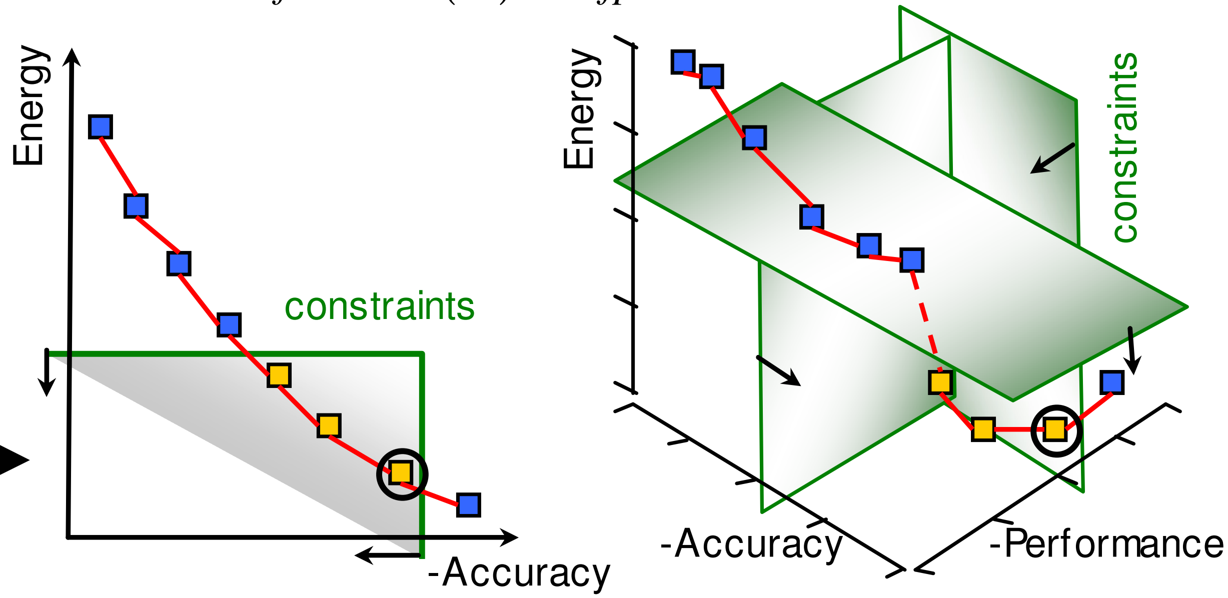
Example: We set constraints on all the three variables. The feasible set is represented by the golden points. We prioritize energy consumption, so our selected realization is the one that also minimizes energy consumption.

The previous problem could be cast as the following optimization problem:

$$\min_{Ri} \ Energy(Ri) \quad \text{subject to}: \begin{array}{l} Accuracy(Ri) \geq 50dB \\ Performance(Ri) \geq 30\,fps \end{array}$$

Circled point: Realization from the feasible set that minimizes energy consumption.

If we ignore one variable (say, performance), we have a 2D optimization problem.



Digital signal, image, and video processing applications

The following systems are discussed:

- *Pixel Processor and Dynamic EPA Management*
- *1D FIR Filter*
- *2D Separable FIR Filter/ Filterbank & Dynamic EPA Management for the 2D FIR Filter*

General Implementation Details

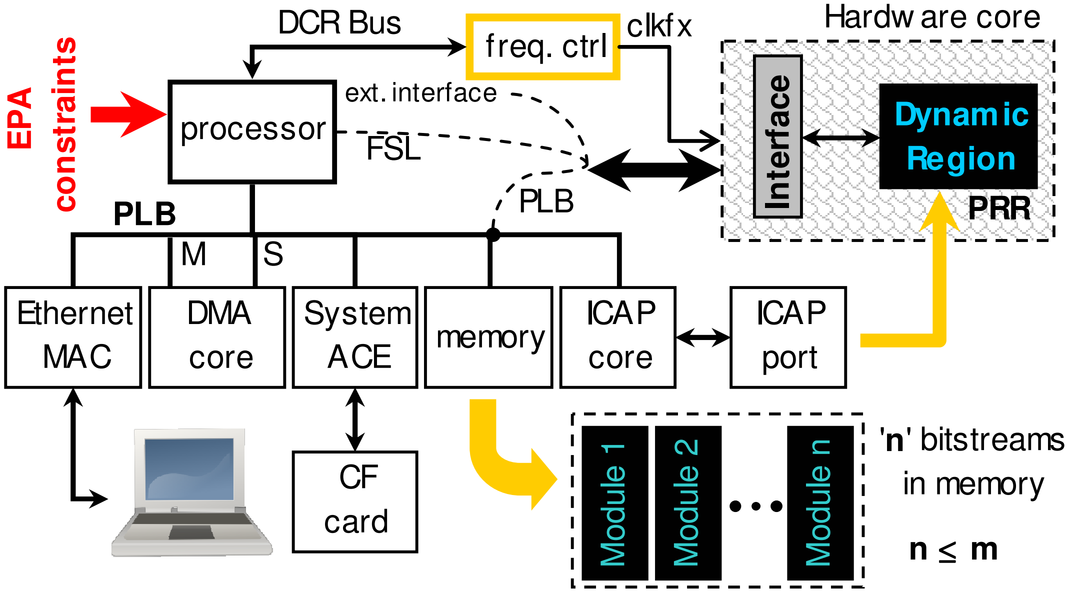
Embedded FPGA system that supports Dynamic Partial Reconfiguration and Dynamic Frequency Control:

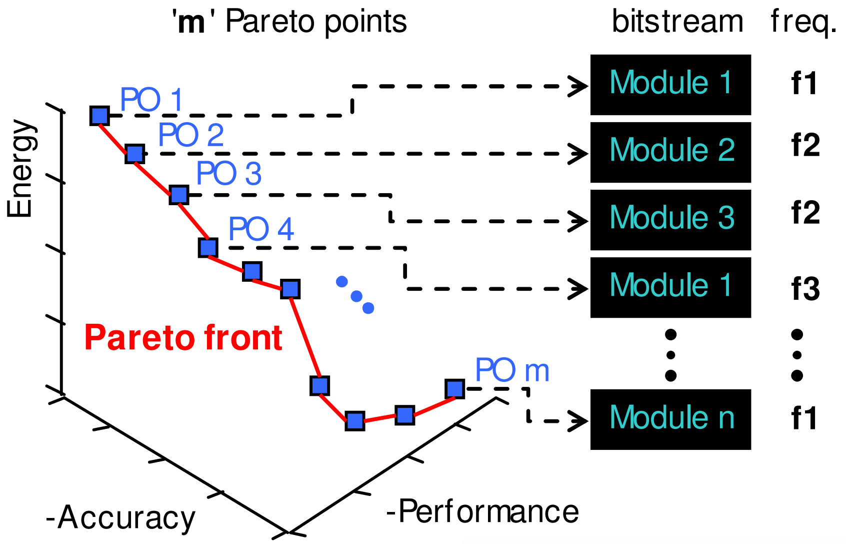
Pareto-optimal point: Represented by <bitstream, frequency of operation>

It is a hardware realization that becomes active in the FPGA via Dynamic Partial Reconfiguration (DPR) and/or Dynamic Frequency Control.

System receives an EPA constraint:

- *It looks for a solution in the Pareto-optimal set: <bitsream*, freq*>*
- *It reconfigures FPGA dynamic region and /or frequency of operation, so as to meet the EPA constraint.*

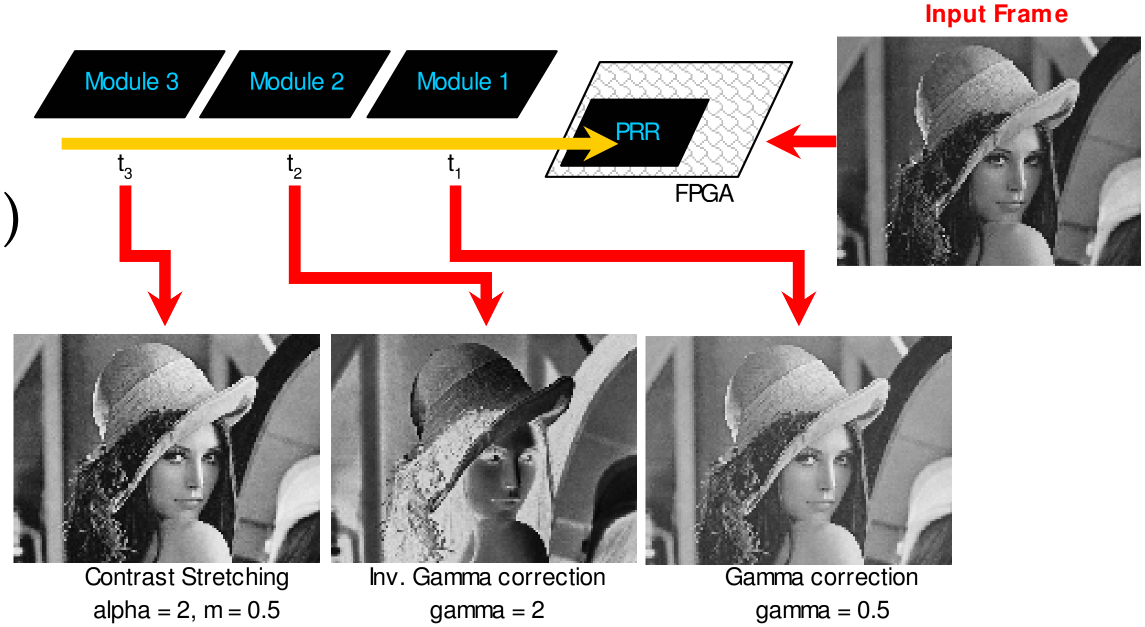


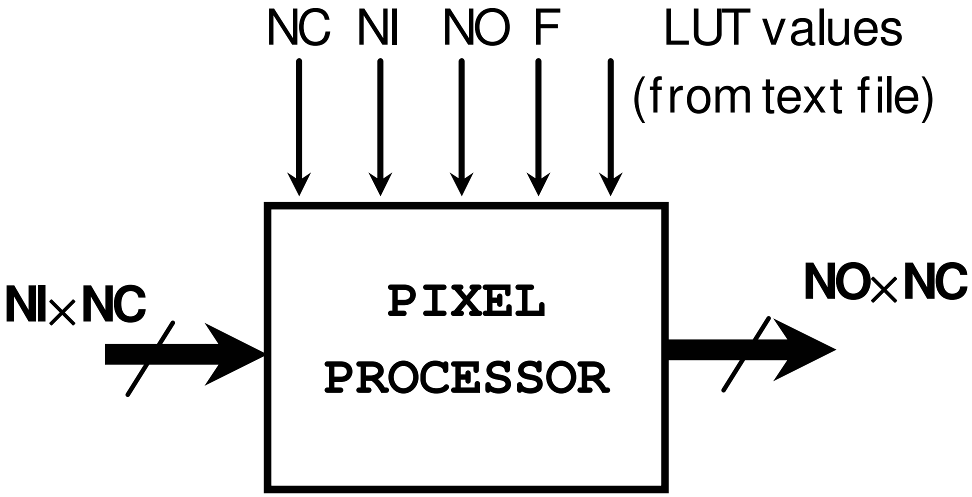


Pixel Processor (1/9)

- Single-pixel operations (e.g., *gamma correction, Huffman encoding, histogram equalization, contrast stretching*) can be dynamically swapped. Parameter *F* modifies the function.

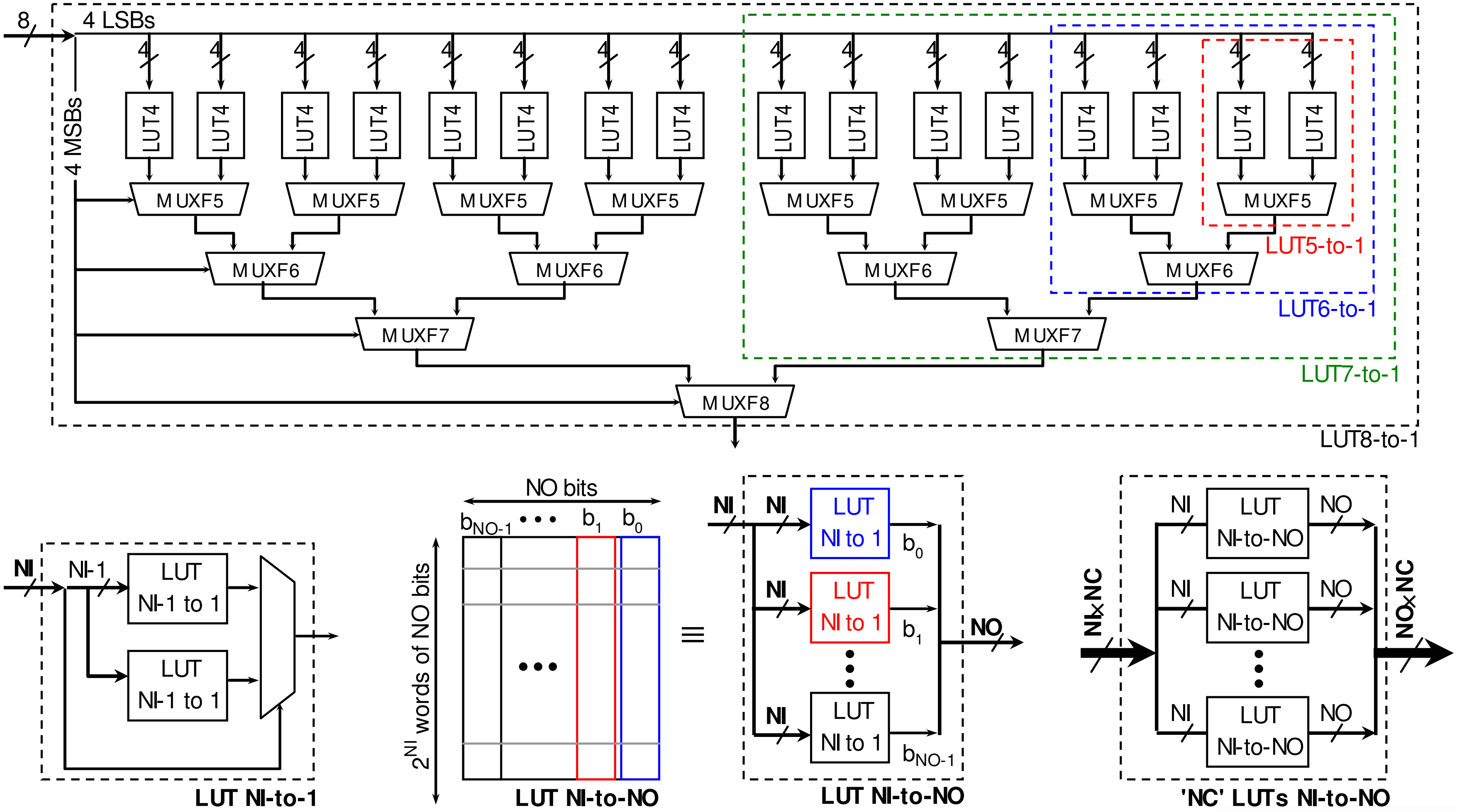
- In addition to dynamically modifying the input-output function, we might want to change:
  - Input pixel bitwidth (*NI*)
  - Output pixel bitwidth (*NO*),
  - Number of parallel processing elements (*NC*)





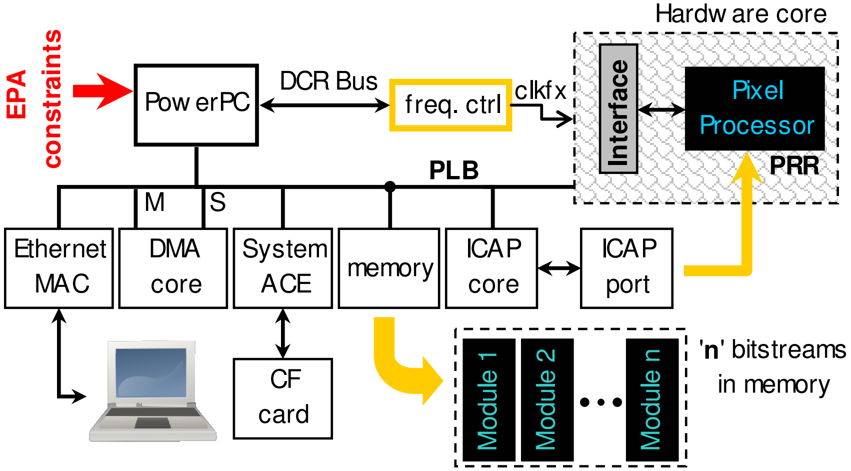
Pixel Processor (2/9)

- **LUT-based architecture**: LUT4 (Virtex-4). LUT6 (Virtex-5, Virtex-6) Up to LUT8-to-1 can be implemented efficiently with Xilinx primitives. For LUT inputs > 8, a recursive implementation is employed.

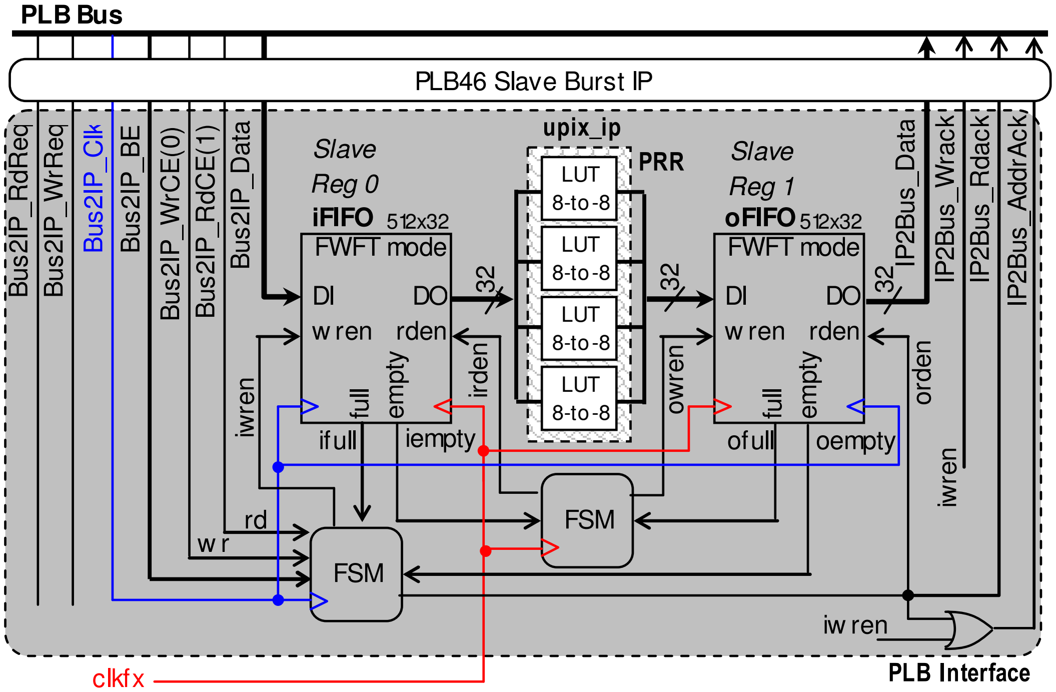


Pixel Processor (3/9)

- **Embedded System**: We create a PLB slave burst interface around the pixel processor core. The figure shows a PRR with NC=4, NI=NO=8.
- The system dynamically reconfigures: NC, NI, NO, FUNCTION, under the following constraints: NI×NC≤32, and NO×NC ≤ 32
- Five 'clkfx' frequencies allowed: 100.00, 66.66, 50.0, 40.00, and 33.33 MHz.
- FIFOs are required to properly isolate different clock regions (PLB clock= 100 MHz and 'clkfx')



Implemented on a ML405 Development Board
Device: XC4VFX20-11FF672



Pixel Processor (4/9)

- **Experimental Setup**: The Pixel Processor is tested under 3 different scenarios. Performance and energy are measured for the IP core.
- Test images: 8-bit '*lena*', 12-bit '*oilp*'
- <u>*Scenario A*</u> (implemented on the embedded system): 32-bit I/O constrained cases. 5 frequencies considered. Parameter *NC* not independent.
- <u>*Scenario B*</u>: 8/12-bit fixed input pixel cases. 5 frequencies considered.
- <u>*Scenario C*</u>: Fixed-frequency constrained implementation. Fixed frequency (100 MHz).

Scenario A

NI	NO (NC)							
5	5(4)	6 4)	7(4)	8(4)	9(2)	10(2)	11(2)	12(2)
6		6(4)	7(4)	8(4)	9(2)	10(2)	11(2)	12(2)
7			7(4)	8(4)	9(2)	10(2)	11(2)	12(2)
8				8(4)	9(2)	10(2)	11(2)	12(2)
9	9(2)	10(2)	11(2)	12(2)	13(2)	14(2)	15(2)	16(2)
10		10(2)	11(2)	12(2)	13(2)	14(2)	15(2)	16(2)
11			11(2)	12(2)	13(2)	14(2)	15(2)	16(2)
12				12(2)	13(2)	14(2)	15(2)	16(2)

Scenario B

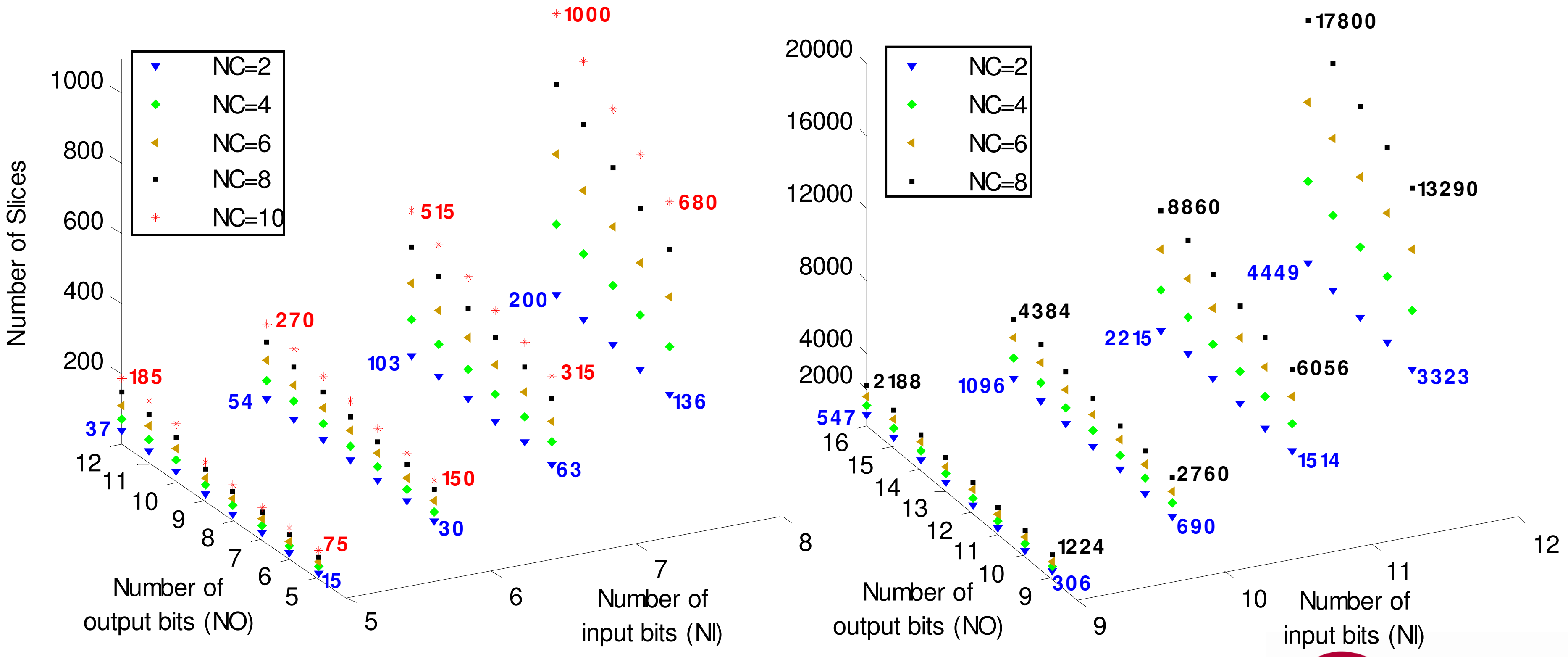
Image	NI	NC	NO				
8-bit	8	2 4 6 8 10	8	9	10	11	12
12-bit	12	2 4 6 8	12	13	14	15	16

Scenario C

Image	NI	NO								NC
8-bit	5	5	6	7	8	9	10	11	12	2
	6		6	7	8	9	10	11	12	4
	7			7	8	9	10	11	12	6
	8				8	9	10	11	12	8
										10
12-bit	9	9	10	11	12	13	14	15	16	2
	10		10	11	12	13	14	15	16	4
	11			11	12	13	14	15	16	6
	12				12	13	14	15	16	8

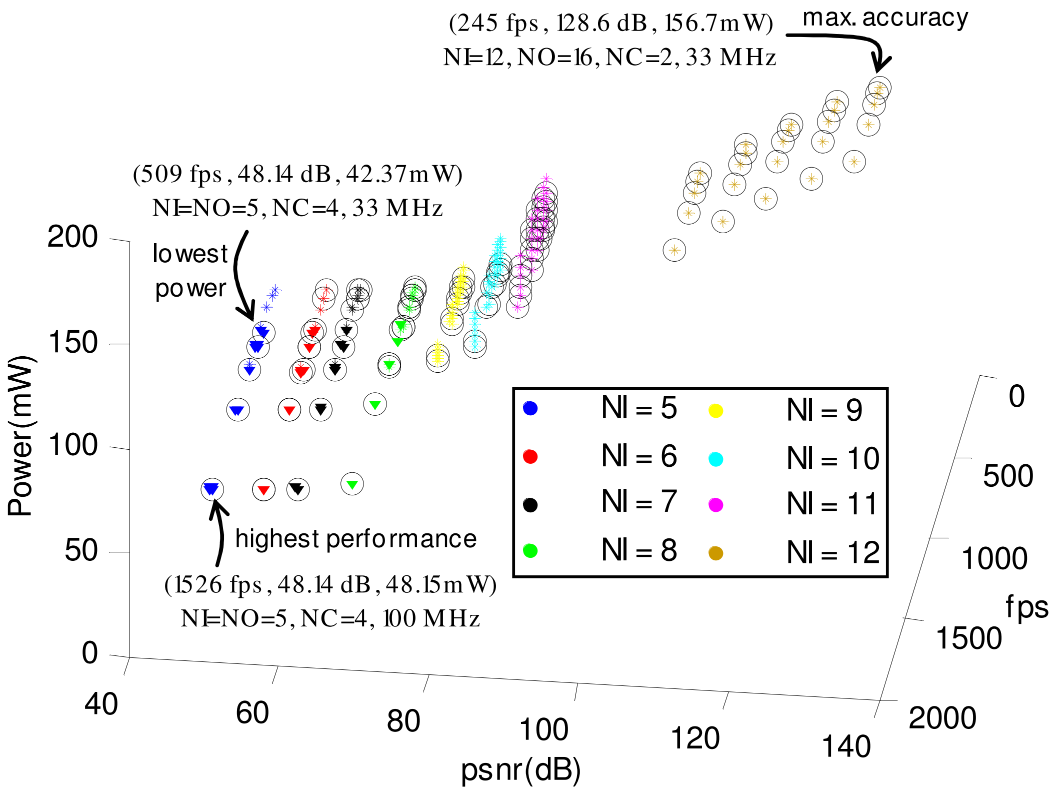
Pixel Processor (5/9)

- **Resource scalability**: Use of Virtex-4 XC4VFX60 FPGA device (25280 Slices) to account for the largest pixel processor realizations. The cases listed in Scenario C are considered (frequency does not vary resources).
- Resource consumption (a function of NI, NO, and NC) grows exponentially with NI, linearly with NC and NO. In the figure, the results are clearly clustered for NI and NC.
- For NI>10 the resource requirements become suboptimal.



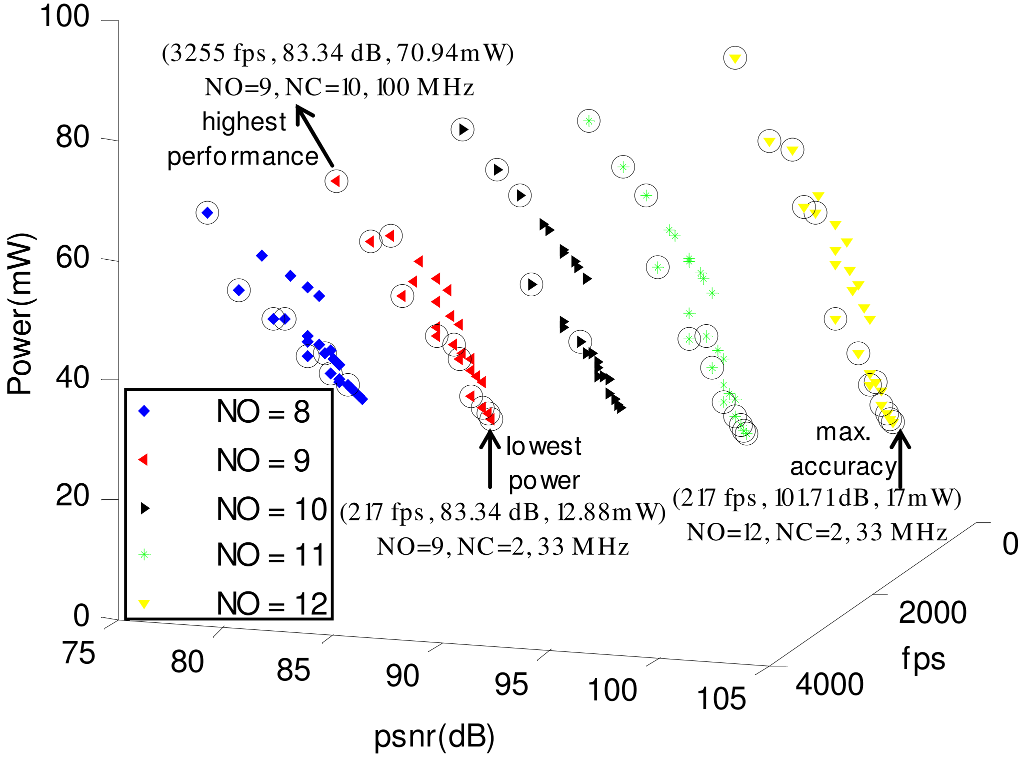
Pixel Processor (6/9)

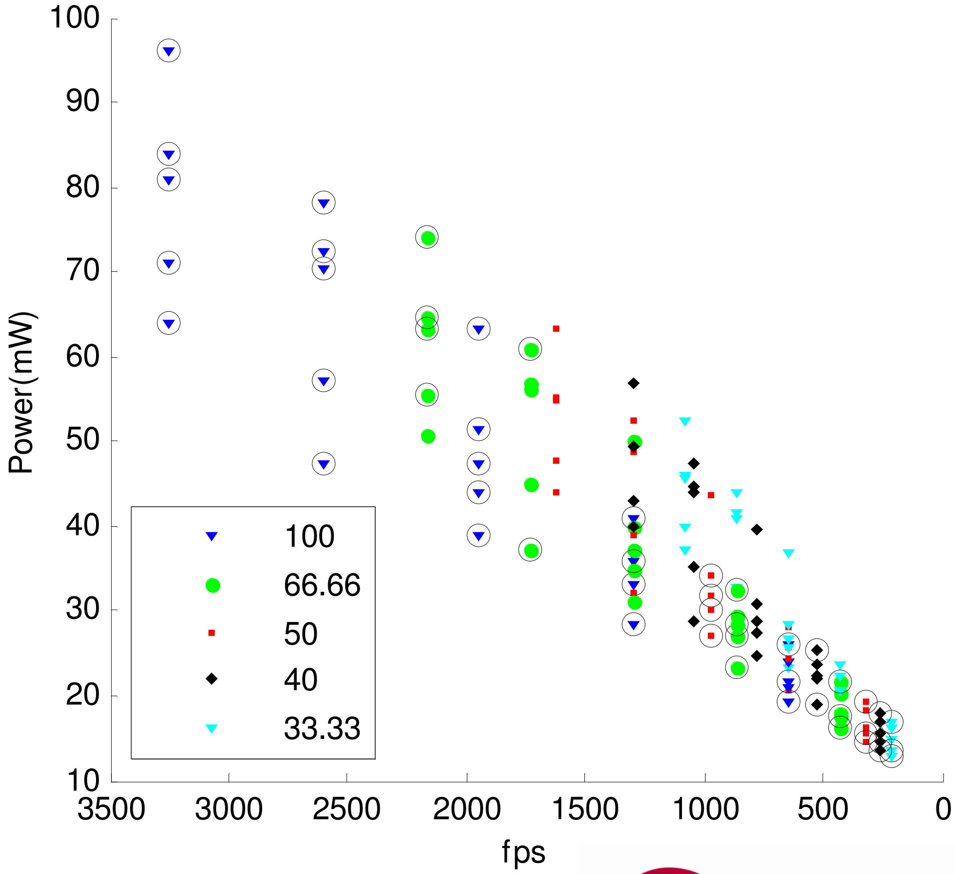
- **Multi-objective optimization of the EPA/PPA space**:
- Gamma correction function (γ =0.5). Power is presented for Scenarios A and B, and energy per frame for Scenario C.
- ***Scenario A***: 12-bit image (NI:12→5). Pareto points cover a wide range of the PPA space (43%) → the approach is effective in generating varied Pareto points.



Pixel Processor (7/9)

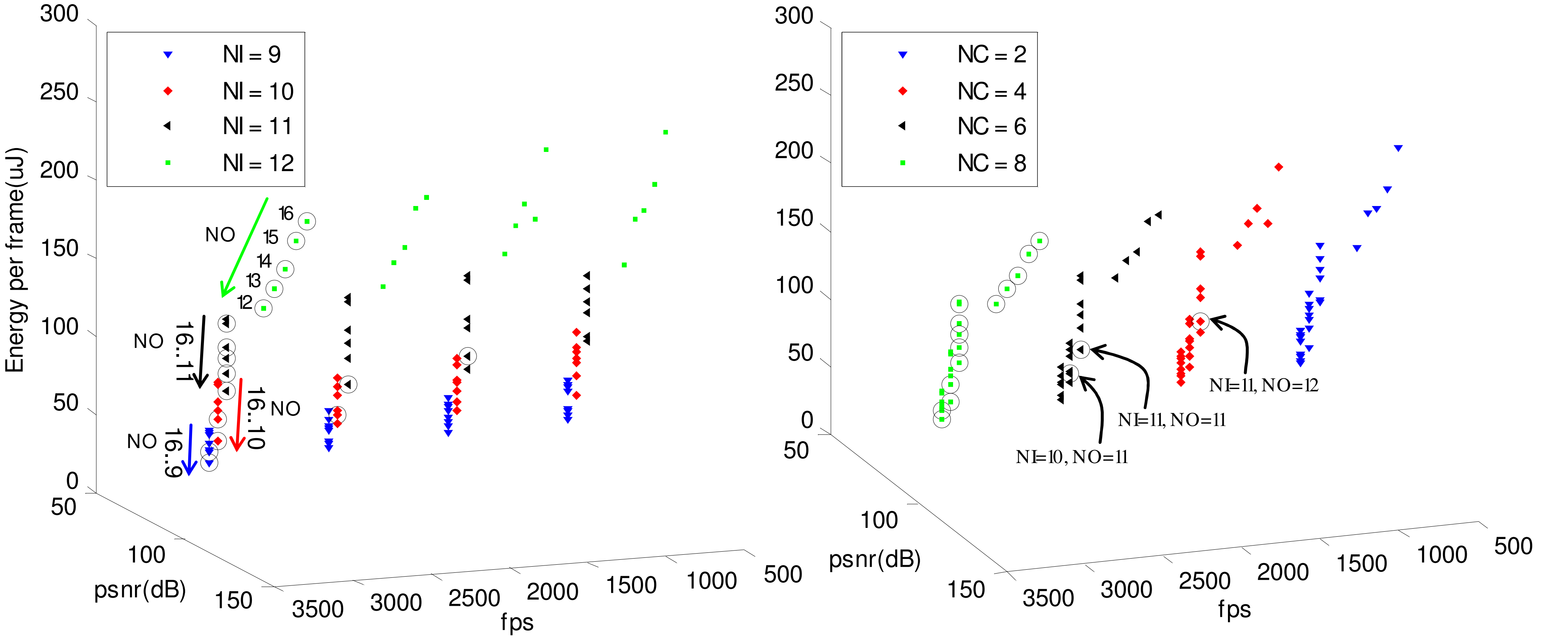
- **Multi-objective optimization of the PPA space:**
- ***Scenario B***: 8-bit input image (*NI=8* fixed). Pareto points are clustered as a function of NO. A similar trend occurs with NC. (not shown)
- Left side shows how power and performance depend on frequency





Pixel Processor (8/9)

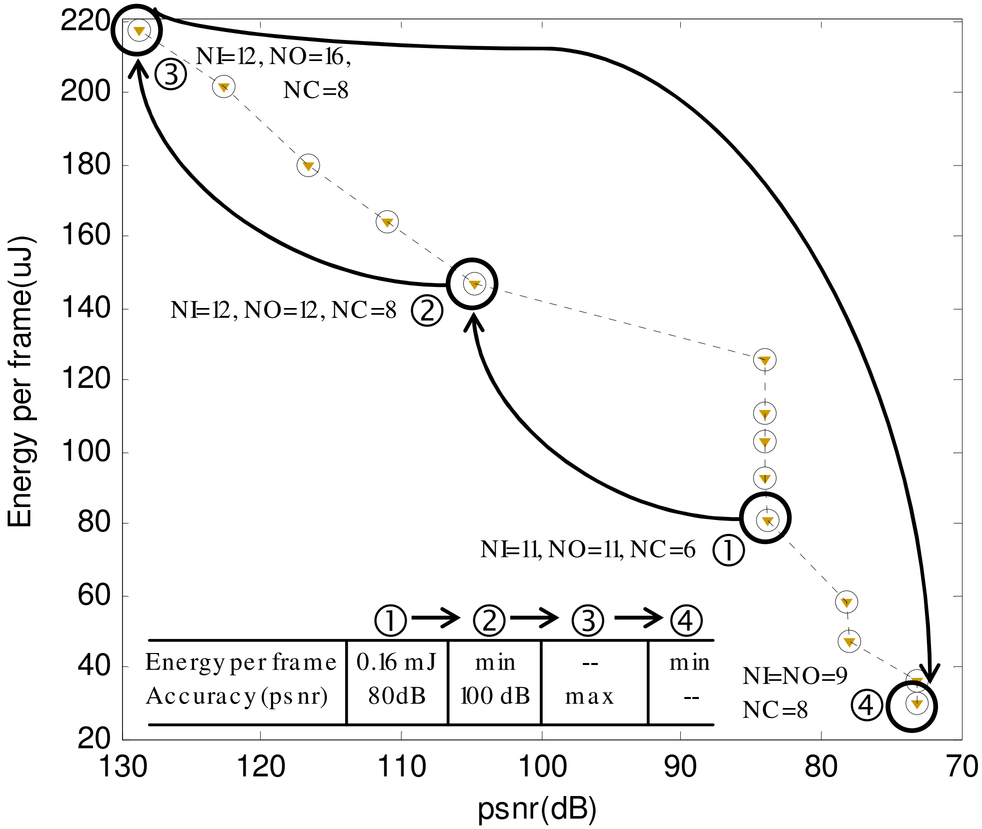
- **Multi-objective optimization of the EPA space:**
- ***Scenario C***: 12-bit input image (NI:12→9), fixed frequency = 100 MHz.
- Performance clusters are defined in terms of the number of cores (NC)
- Energy clusters are defined in terms of the input pixel bitwidth (NI)



Pixel Processor (9/9)

- **Dynamic EPA management**: We show an example on 2D (ignoring performance) with time-varying constraints:

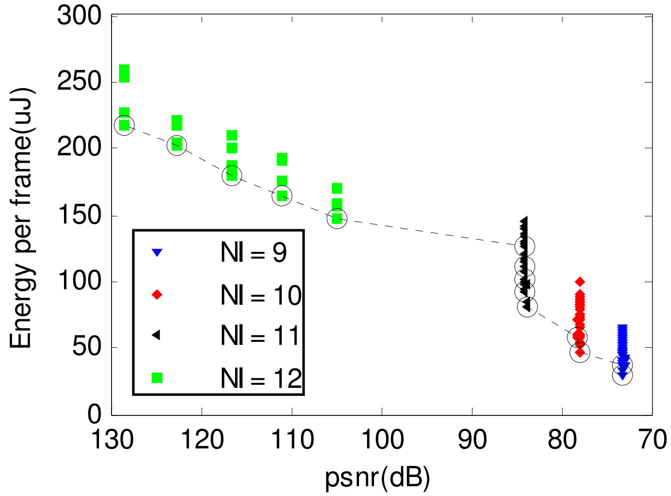
1. Require accuracy≥80dB and Energy ≤160uJ.
2. Minimize energy subject to Accuracy ≥ 100dB
3. Maximize Accuracy.
4. Minimize Energy consumption.



	①	②	③	④
Energy per frame	0.16 mJ	min	--	min
Accuracy (psnr)	80dB	100 dB	max	--

FIXED-FREQUENCY (100 MHz) CONSTRAINED IMPLEMENTATIONS: PARETO OPTIMAL POINTS (12-BIT IMAGE)

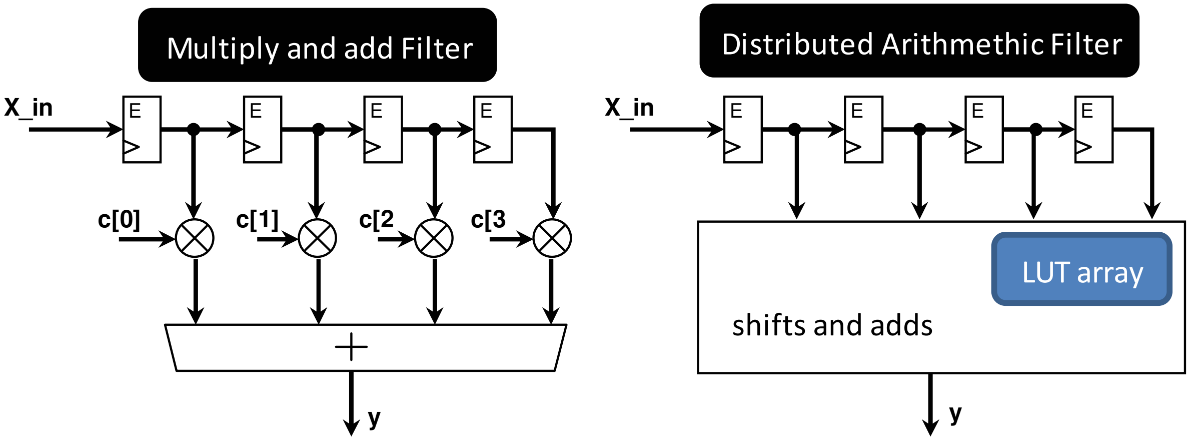
NI	NO	NC	psnr(dB)	Energy per frame (uJ)
9	9	8	73.1611	29.4850
9	11	8	73.1667	36.0877
10	10	8	77.9215	46.6695
10	11	6	78.0665	57.8377
11	11	6	83.8819	81.0746
11	12	4	83.9695	92.4202
11	13	8	83.9751	102.4708
11	14	8	83.9875	110.5668
11	15	8	83.9922	125.3556
12	12	8	104.7546	146.9356
12	13	8	110.8823	163.8397
12	14	8	116.6600	179.2773
12	15	8	122.6959	201.4623
12	16	8	128.5966	217.2102



* Submitted to IEEE Transactions on Circuits and Systems for Video Technology

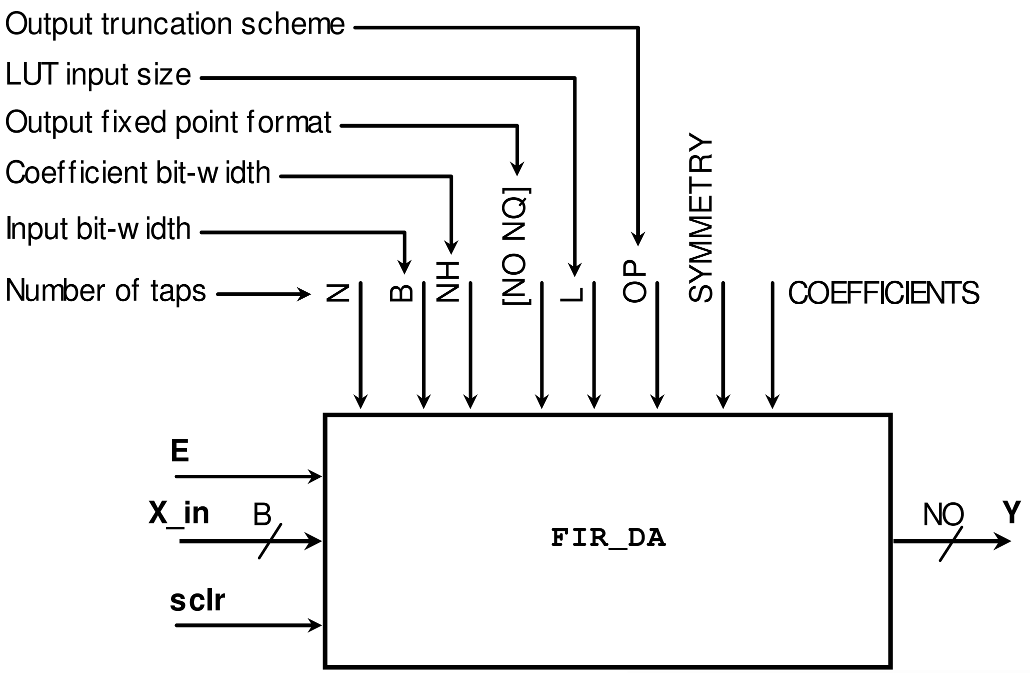
1D FIR Filter (1/3)

TWO TYPES OF IMPLEMENTATIONS



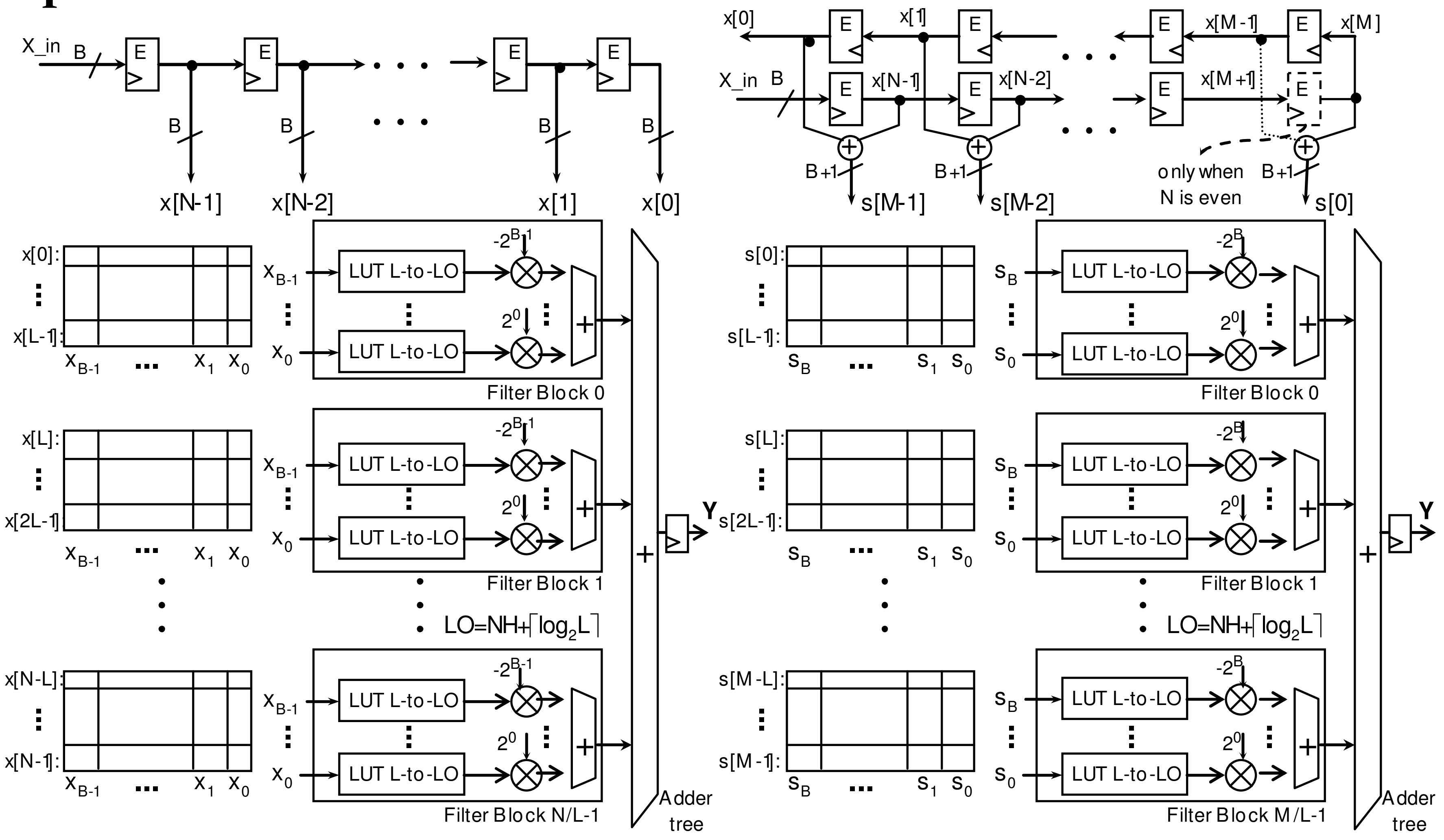
Distributed Arithmetic (DA) approach is more efficient since it is a LUT-based approach that turns the multiplications into shifts and adds. But it requires the coefficients to be constant.

- **Efficient implementation of a 1D FIR Filter via DPR:** Dynamic Partial Reconfiguration turns the fixed-coefficient DA filter into a variable-coefficient DA filter, at the expense of partial reconfiguration time overhead.
- **Parameterization of the VHDL-coded FIR filter core:**



1D FIR Filter (2/3)

- **LUT-based architecture for the Distributed Arithmetic Implementation:**



1D FIR Filter (3/3)

- **Filter Block Implementation →**

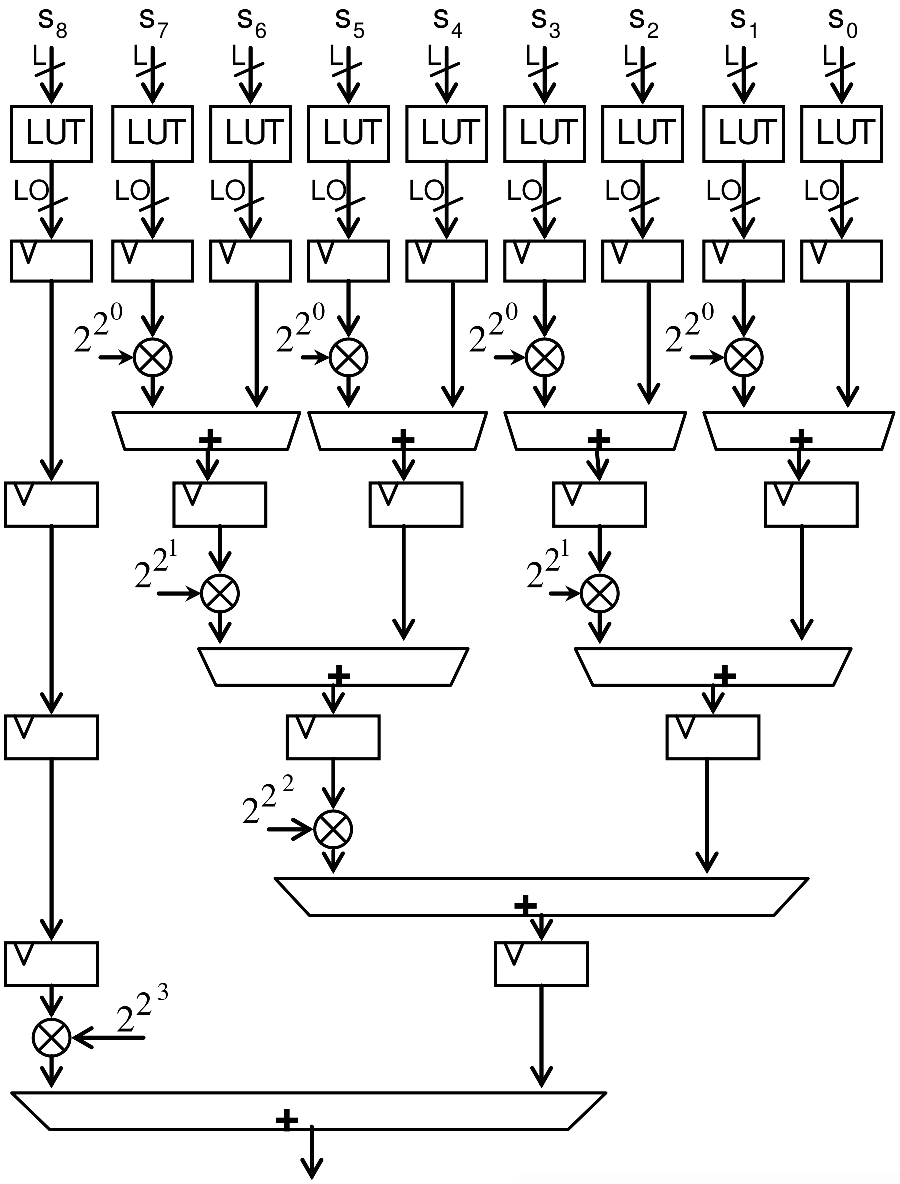
- **Two implementations:**

  1) Coefficient-only reconfiguration: Only the coefficient values can be dynamically modified (PRR is made of the set of LUTs)

  2) *Full-filter reconfiguration*: It allows the run-time modification of all parameters.

- Performance dependence as the reconfiguration rate increases was shown.

* This work was published in the 2009 International Journal of Reconfigurable Computing


Filter Block Implementation (Symmetric, B=8)

2D FIR Filter (1/10)

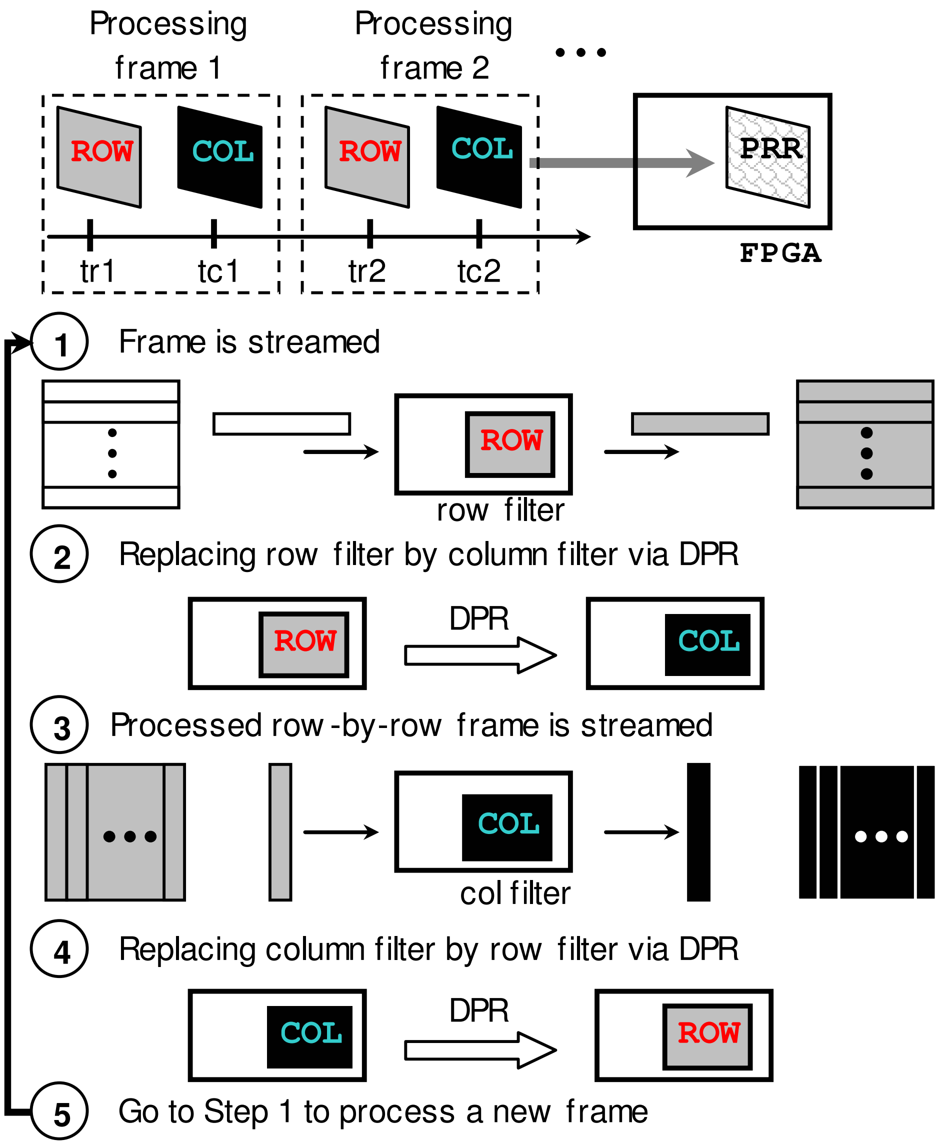
2D Separable Filter Implementation:

- Separable FIR filters allow for efficient implementations by means of two 1D FIR Filters.
- The reconfiguration rate is constant (twice per frame).
- Cyclic Dynamic reconfiguration of two 1-D filters (usually full-filter reconfiguration):

  - Implement row filter
  - Replace by column filter
  - Implement column filter
  - Replace by row filter

  …

* A comparison of this 2D FIR Filter and a GPU implementation for different number of coefficients was published 2011 IEEE Field Programmable Logic Conference (FPL'2011)



1 Frame is streamed

2 Replacing row filter by column filter via DPR

3 Processed row-by-row frame is streamed

4 Replacing column filter by row filter via DPR

5 Go to Step 1 to process a new frame

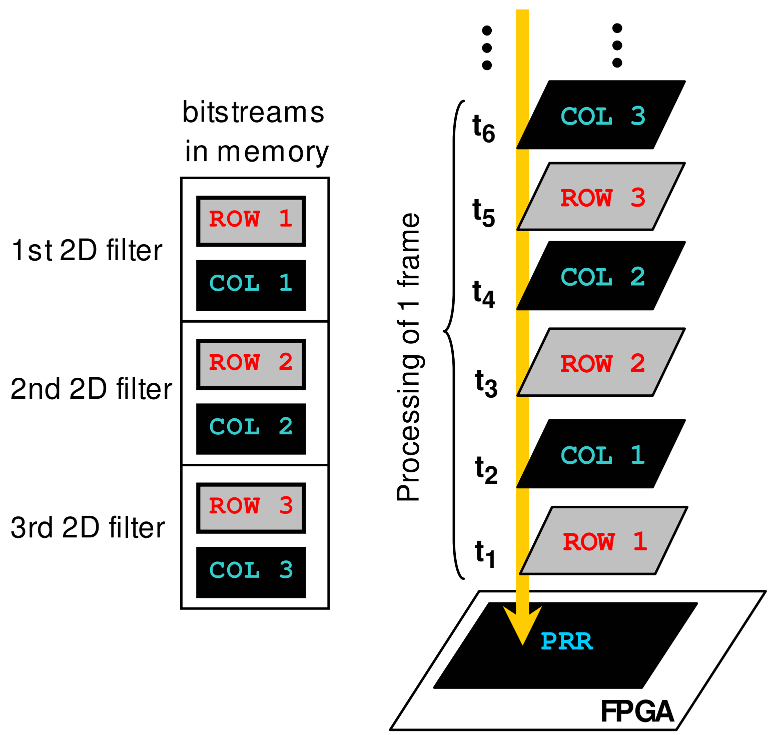
2D FIR Filter (2/10)

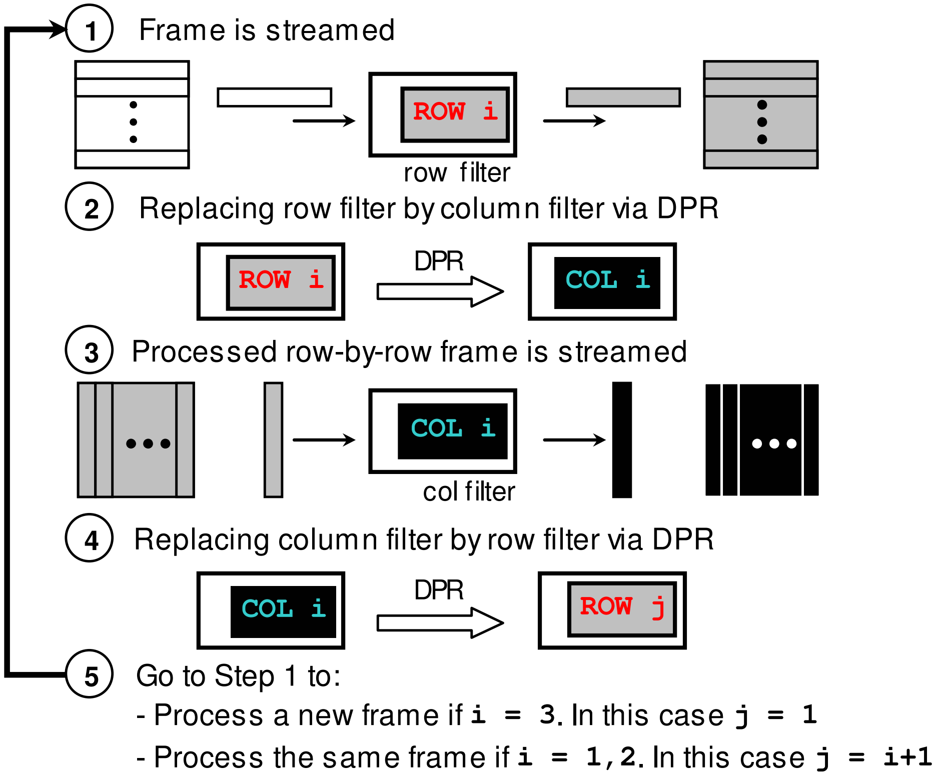
2D Separable Filterbank Implementation:

Simple modifications to 2-D filter implementation:

- Reconfigure with the next 2-D filter and re-process frame
- When all filters have been applied, move to the next-frame and back to the first 2-D filter

EXAMPLE: 3-CHANNEL 2-D FILTERBANK



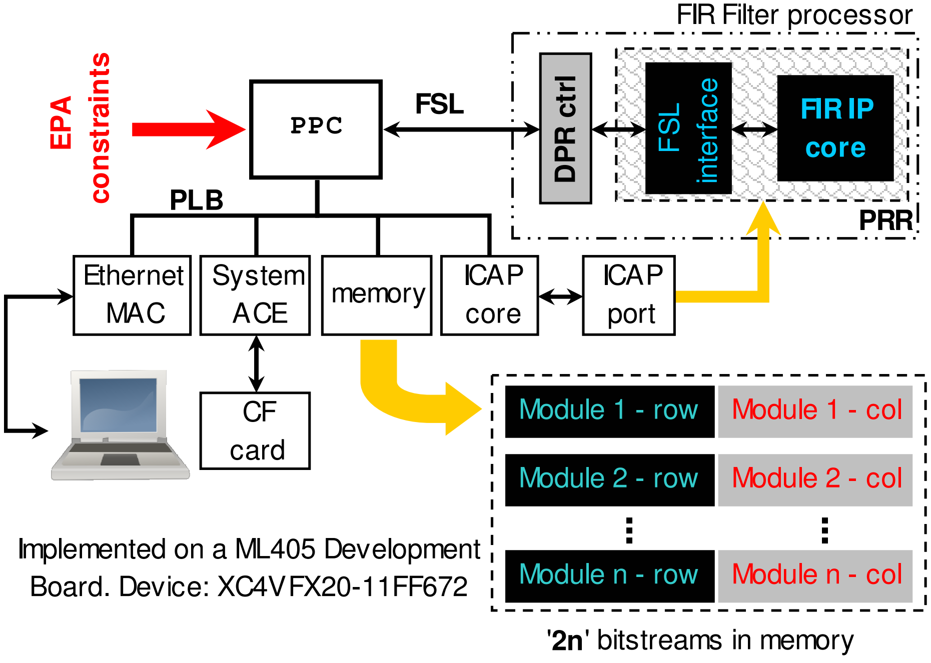


1. Frame is streamed
2. Replacing row filter by column filter via DPR
3. Processed row-by-row frame is streamed
4. Replacing column filter by row filter via DPR
5. Go to Step 1 to:
   - Process a new frame if `i = 3`. In this case `j = 1`
   - Process the same frame if `i = 1,2`. In this case `j = i+1`

* This work was published in the in the 2010 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI 2010)

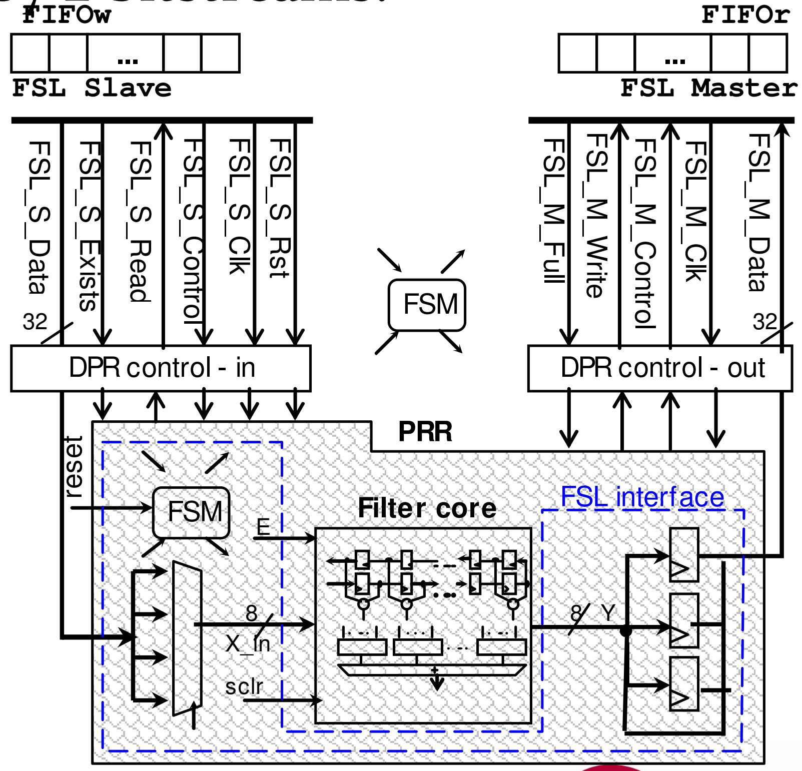
2D FIR Filter (3/10)

- **Embedded System**: We create a FSL interface around the FIR filter core. The *full-filter reconfiguration* core is considered since it let us vary all the filter parameters, thereby allowing the creation of a large EPA space.
- FSL interface: We included this interface inside the PRR, so that we dynamically modify the I/O bitwidth.
- DPR control block: disables the PRR outputs during reconfiguration and resets the flip flops of the PRR after each partial reconfiguration.
- Each 2D filter realization is represented by 2 bitstreams.



Implemented on a ML405 Development Board. Device: XC4VFX20-11FF672

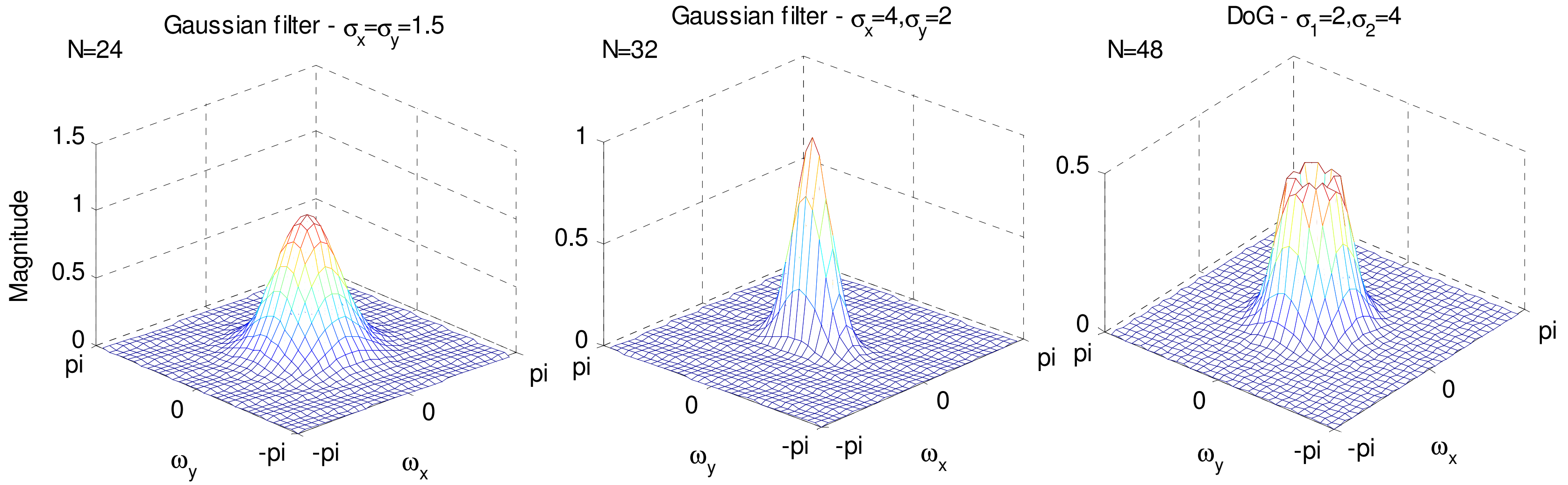
'**2n**' bitstreams in memory



2D FIR Filter (4/10)

Experimental Setup: The FIR core is fully parameterized. The table describes the combination of parameters we utilize that creates the Energy-Performance-Precision (EPA) space.

- Performance (fps) and energy are measured for the IP core.
- Test image: 8-bit ‘*lena*’ (VGA, CIF, QCIF frame sizes).
- Three different Gaussian filters:



Frame Size	Number of coefficients (N)						Coefficient bit-width (NH)	Output bitwidth (OB)
640x480 (VGA) 352x288 (CIF) 176x144 (QCIF)	8	12	16	20	24	32	10 12 16	8 16

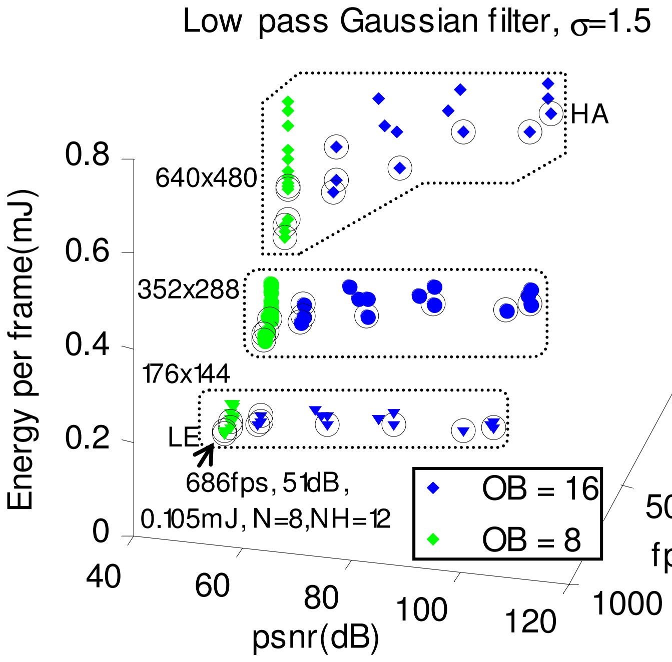
2D FIR Filter (5/10)

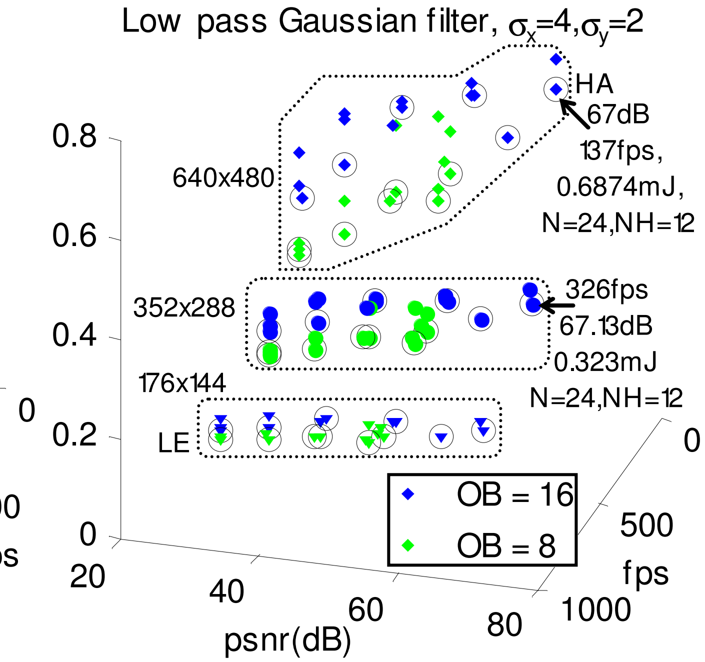
Multi-objective optimization of the EPA space:

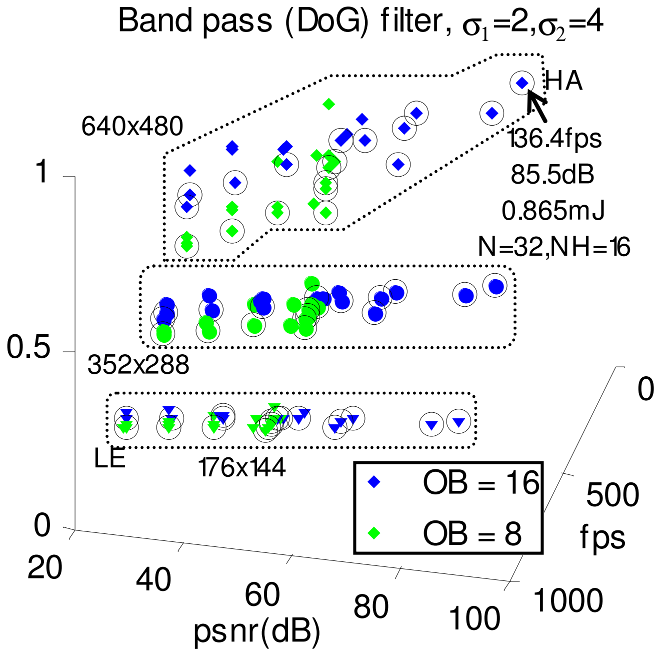
Results for the 3 filters and 3 image sizes: The highest accuracy is achieved by increasing the number of coefficients (*N*), the coefficient bitwidth (*NH*), and with 16 output bits (*OB*). Frame size increases energy per frame. Thus, we present Pareto-optimal results independently of the frame size.

HA: highest accuracy realization from the Pareto front

LE: lowest energy realization from the Pareto front.


Low pass Gaussian filter, σ=1.5


Low pass Gaussian filter, σ_x=4,σ_y=2

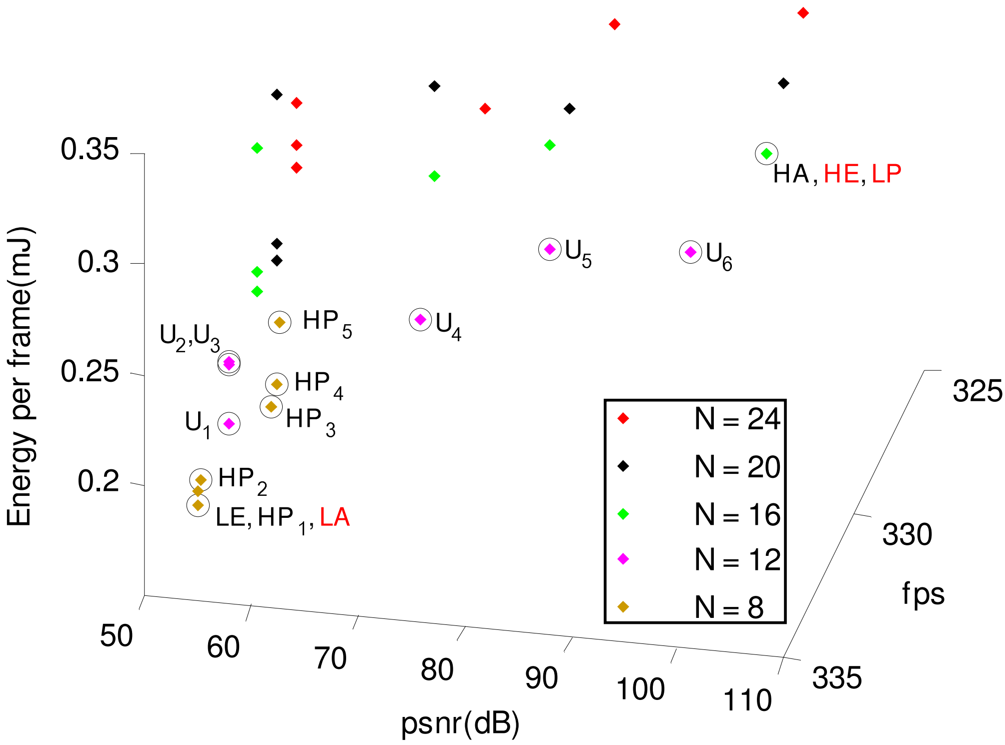

Band pass (DoG) filter, σ_1=2,σ_2=4

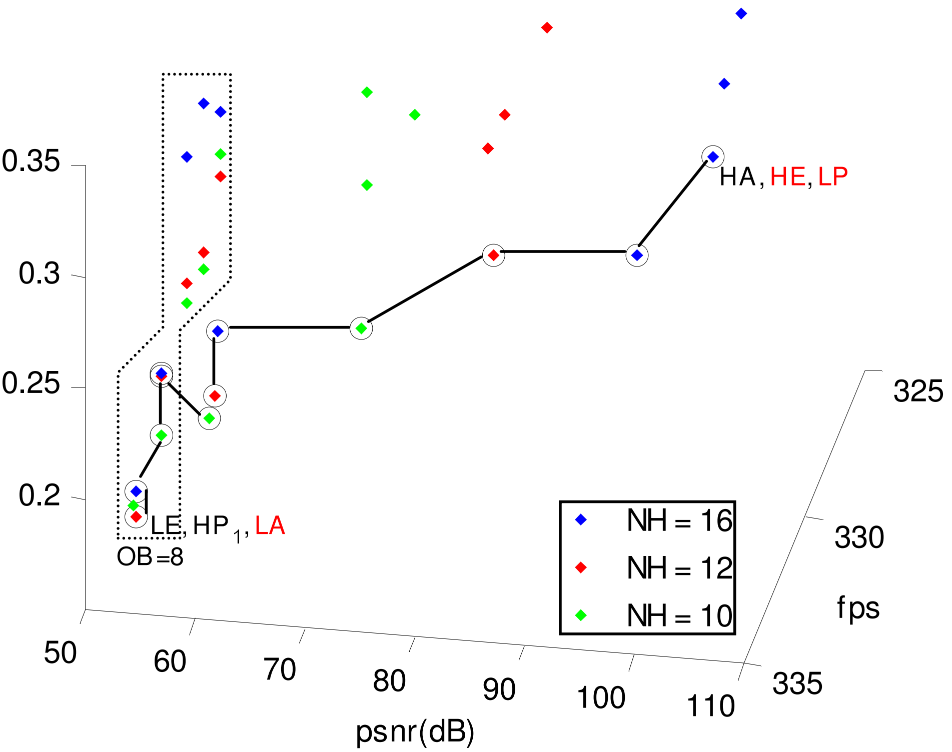
2D FIR Filter (6/10)

Multi-objective optimization of the EPA space:

Results for the 3 filters and CIF frame size: We show the Pareto-optimal realizations as a function of *N*, *NH*, and *OB*.

Low-pass Gaussian Filter, σx=σy=1.5:

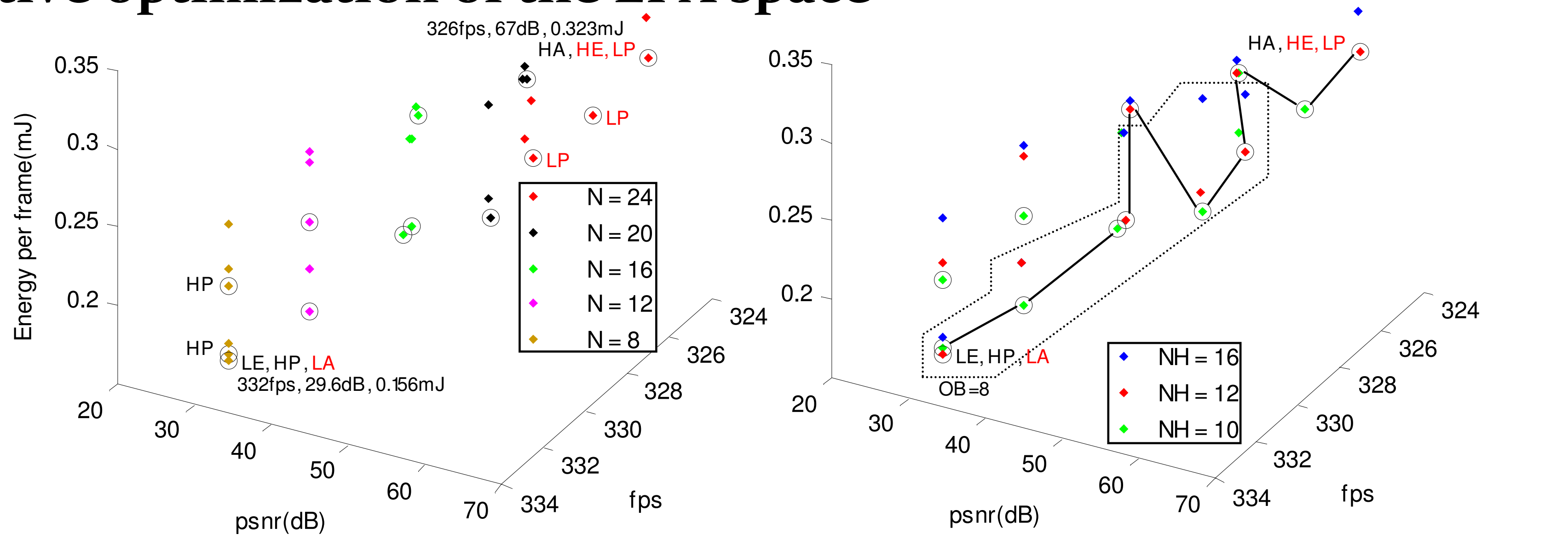




2D FIR Filter (7/10)

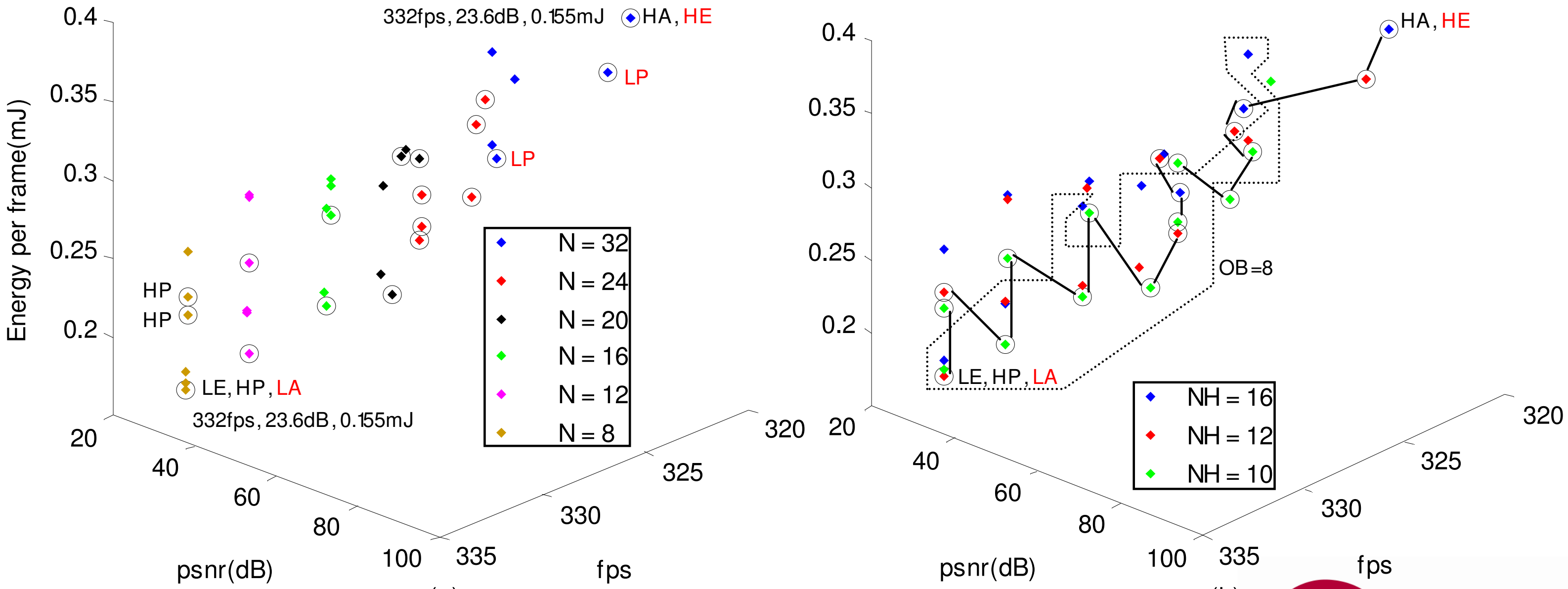
Multi-objective optimization of the EPA space

Low-pass Gaussian Filter, $\sigma x=4,\sigma y=2$



Difference of Gaussians (DoG) filter

$\sigma 1=2,\sigma 2=4$



(a)

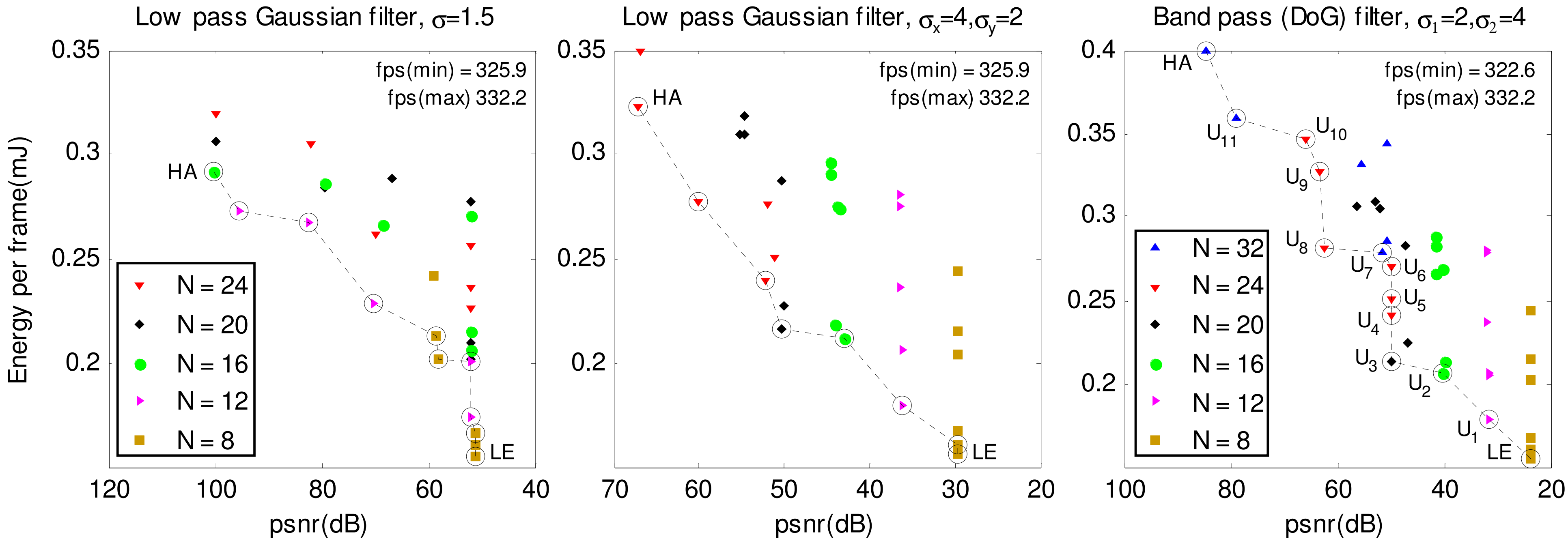
(b)

2D FIR Filter (8/10)

Multi-objective optimization of the <u>Energy-Accuracy</u> space:

Performance results are over 100 fps (VGA), and 300 fps (CIF). Overall, for a fixed frame size, performance does not vary significantly. Thus, it makes sense to restrict our attention to the Energy-Accuracy Space

Results for the 3 filters and CIF frame size: We show the Pareto-optimal realizations as a function of *N* (number of coefficients).



2D FIR Filter (9/10)

Dynamic EPA Management (1st example)**:** Applied on the Pareto front of the DoG filter (see table). Video sequence: 'foreman'.

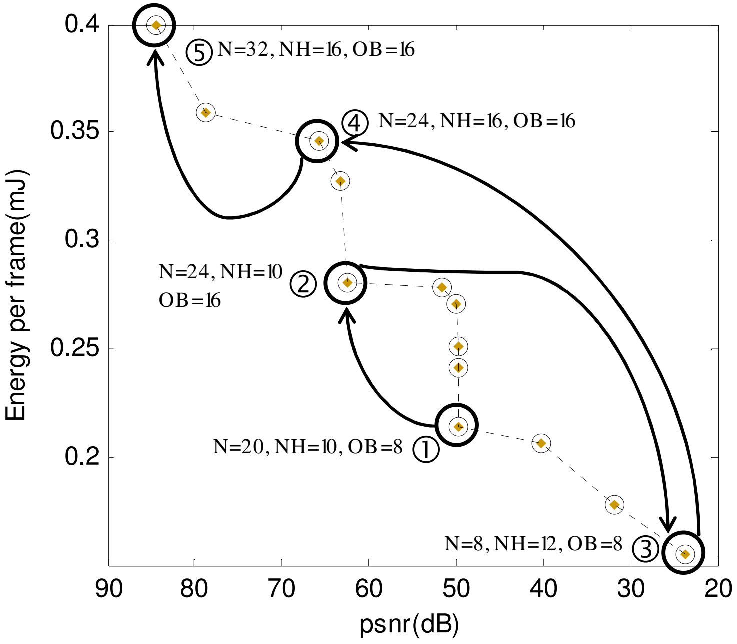
	N	NH	OB	psnr(dB)	Energy per frame (mJ)
LE	8	12	8	23.6345	0.1551
U_1	12	10	8	31.7124	0.1780
U_2	16	10	8	40.2984	0.2062
U_3	20	10	8	49.5758	0.2139
U_4	24	12	8	49.6447	0.2416
U_5	24	10	8	49.8104	0.2508
U_6	24	16	8	49.9967	0.2704
U_7	32	10	8	51.6392	0.2781
U_8	24	10	16	62.3297	0.2807
U_9	24	12	16	63.1992	0.3276
U_{10}	24	16	16	65.8199	0.3464
U_{11}	32	12	16	78.7569	0.3586
HA	32	16	16	84.4371	0.3991

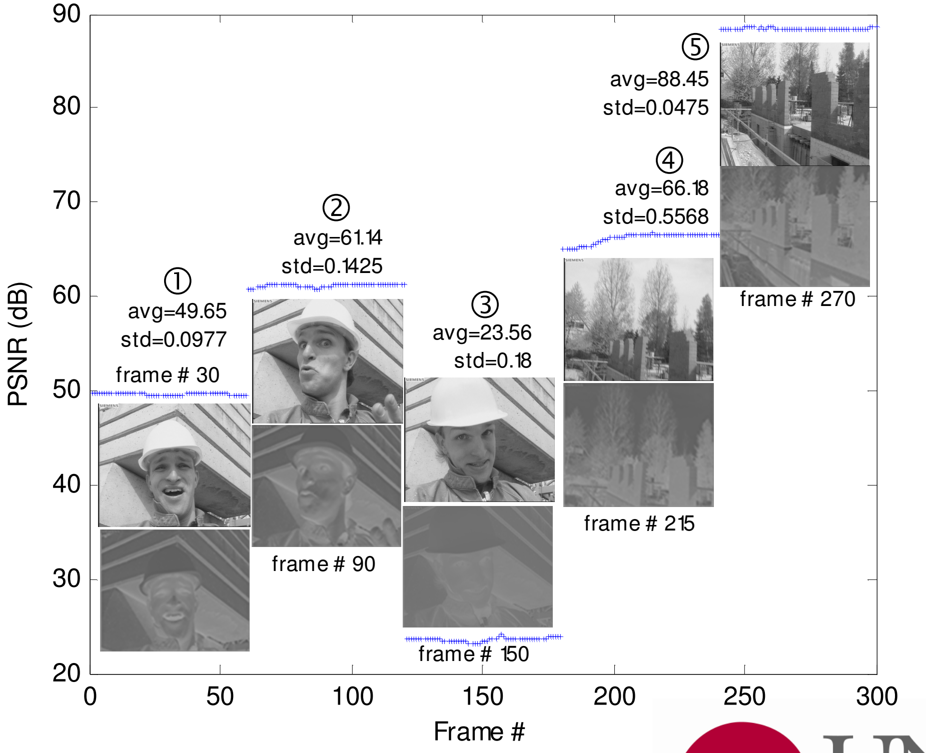
Time-varying sequence of constraints:

1. Require Accuracy ≥ 45dB and Energy ≤ 0.3mJ
2. Minimize Accuracy subject to Energy ≤ 0.3mJ
3. Minimize Energy per frame consumption
4. Minimize Energy subject to Accuracy ≥ 65 dB
5. Maximize Accuracy

① → ② → ③ → ④ → ⑤

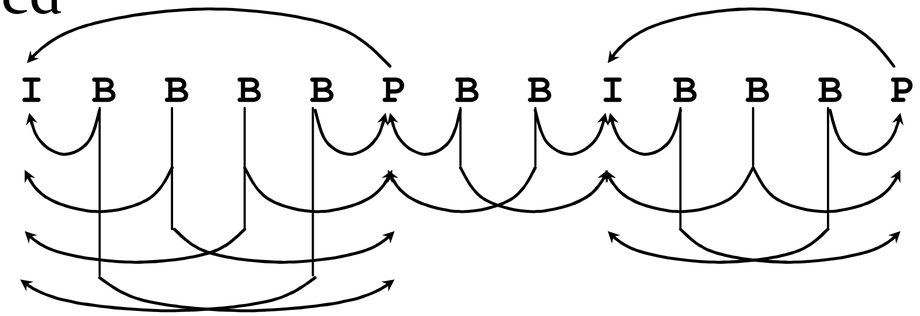
Energy per frame	0.3 mJ	0.3mJ	min	min	--
Accuracy (psnr)	45dB	max	--	65dB	max





2D FIR Filter (10/10)

Dynamic EPA Management (2nd example):
Suppose that the video output will be streamed through a communications channel. Here, it makes sense to impose real-time EPA constraints based on the Group Of Pictures (GOP). The GOP describes the prediction relationships between frame types (MPEG-1 recommendation):



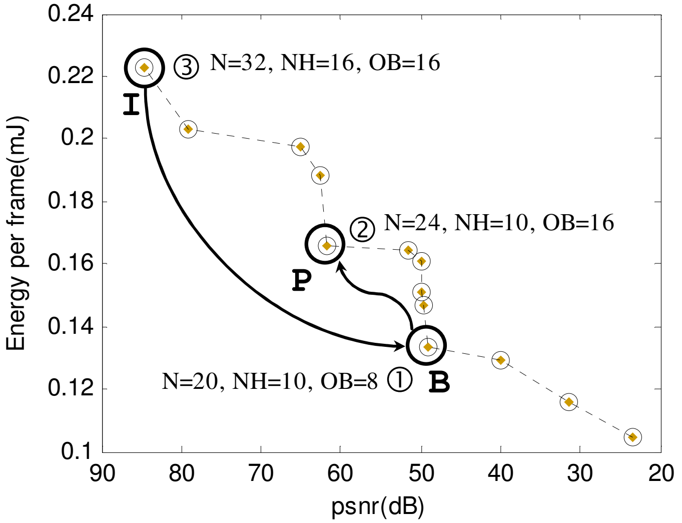
The GOP can now be defined for different videos.

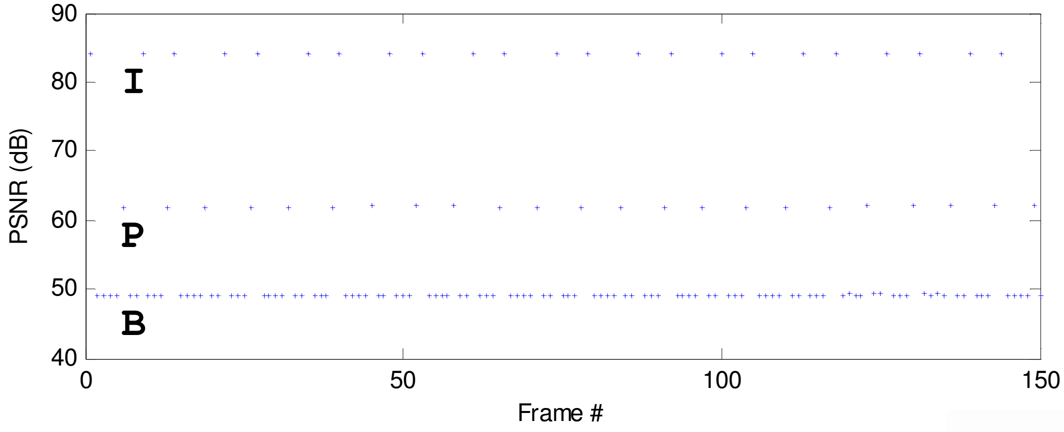
It makes sense to impose the following accuracy constraints:

-I-frames should be of the highest accuracy (all frames depend on it): Point ③

-P-frames should be of very high accuracy (many frames depend on it): Point ②

-B-frames can be of low accuracy (no frames depend on it): Point ①





'missamerica' video sequence (QCIF)

* To be submitted to IEEE Transactions on Image Processing

Conclusions

- *A framework was presented for the generation of optimal realizations (in the multi-objective sense) from the Energy-Performance-Accuracy space. The framework allows for dynamic EPA management for digital signal, image, and video processing applications.*
- *Dynamic EPA management is based on Dynamic Partial Reconfiguration (DPR) and Dynamic Frequency Control to deliver performance with limited hardware resources and relatively low energy consumption.*
- *The framework was tested on a Pixel Processor architecture and a 2D FIR Filtering system. Dynamic EPA management was demonstrated on two standard video sequences.*
- *The results suggest that the general framework can be applied to a variety of digital signal, image, and video processing systems. The framework can be greatly improved by the automatic generation of time-varying constraints (e.g., detection of a scene triggers a requirement for increased accuracy, a scene remaining still triggers a requirement for a decrease in energy consumption)*
- *Ultimately, this framework will lead to exciting new methods that allow for systems to only switch between architectures that are optimal in the multi-objective sense.*

Related publications

[1] D. Llamocca, M. Pattichis, and A. Vera, "A Dynamically Reconfigurable Parallel Pixel Processing System", in *Proceedings of 2009 International Conference on Field Programmable Logic and Applications FPL'2009*, Prague, Czech Republic, Sep. 2009.

[2] D. Llamocca, M. Pattichis, and A. Vera, "A dynamically reconfigurable platform for fixed-point FIR filters," in *Proceedings of the International Conference on Reconfigurable Computing and FPGAs ReConFig'09*, Cancun, Mexico, Dec. 2009.

[3] D. Llamocca, M.S. Pattichis, and G. A. Vera, "A dynamic computing platform for image and video processing applications," in *Proceedings of the 43rd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2009.

[4] G. A. Vera, D. Llamocca, M. S. Pattichis, and J. Lyke, "A dynamically reconfigurable computing model for video processing applications," in *Proc. of the 43rd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2009.

[5] D. Llamocca, M. Pattichis, and G. A. Vera, "Partial Reconfigurable FIR Filtering system using Distributed Arithmetic", *International Journal of Reconfigurable Computing*, vol. 2010, Article ID 357978, 14 pages, 2010.

[6] D. Llamocca, M. Pattichis, "Real-time dynamically reconfigurable 2-D filterbanks", in *Proceedings of 2010 IEEE Southwest Symposium on Image Analysis & Interpretation*, Austin, TX, May. 2010.

[7] D. Llamocca, M.S. Pattichis, G. A. Vera, and J. Lyke, "Dynamic Partial Reconfiguration through Ethernet Link", in *Proceedings of the 2010 AIAA Infotech Conference at Aerospace*, Atlanta, GA, USA, April 2010.

[8]D. Llamocca, C. Carranza, and M. Pattichis, "Separable FIR Filtering in FPGA and GPU implementations: Energy, Performance, and Accuracy considerations", in *Proceedings of 2011 International Conference on Field Programmable Logic and Applications FPL'2011*, Chania, Greece, Sep. 2011.