

Learning Computer Architecture through the ASIP Paradigm: A Research-oriented Approach

Synopsys ASIP University Day
2021-11-17

Pierre Langlois, Eng., Ph.D.
Professor and Chair
Department of Computer and Software Engineering



**POLYTECHNIQUE
MONTRÉAL**

Outline

- About me
- ASIP related research in Polytechnique Montréal
- ASIPs are the ideal abstraction to learn computer architecture
- INF8505: an ASIP design course
- A course project with Synopsys' ASIP Designer
- Flipped-classroom meets ASIP research
- Lessons learned
- About the lightboard

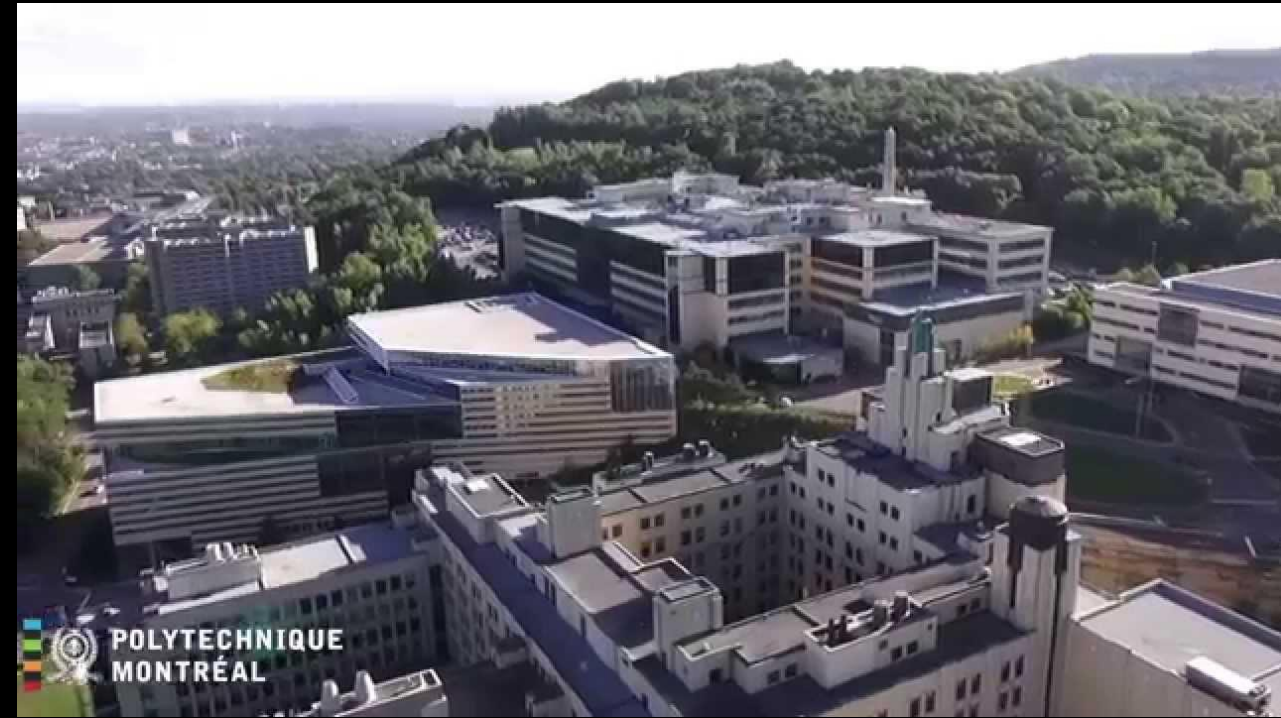
Pierre Langlois, Eng., Ph.D.

- Professor in Polytechnique Montréal since 2005
- Chair of Computer and Software Engineering Department since 2016
- Research interests :
 - Computer architecture and custom processors
 - Network processors, languages and compilation
 - Implementation of artificial intelligence, deep learning and image processing algorithms
- Many industrial collaborations, past and current
- Co-author or author of ~110 refereed papers
 - ~1400 citations | h-index 19
- Principal or co-advisor of ~40 research and ~45 non-research graduate students since 2002
- Teaching of various computer architecture courses, FPGAs, DSP, etc.

- <https://www.polymtl.ca/expertises/langlois-pierre>

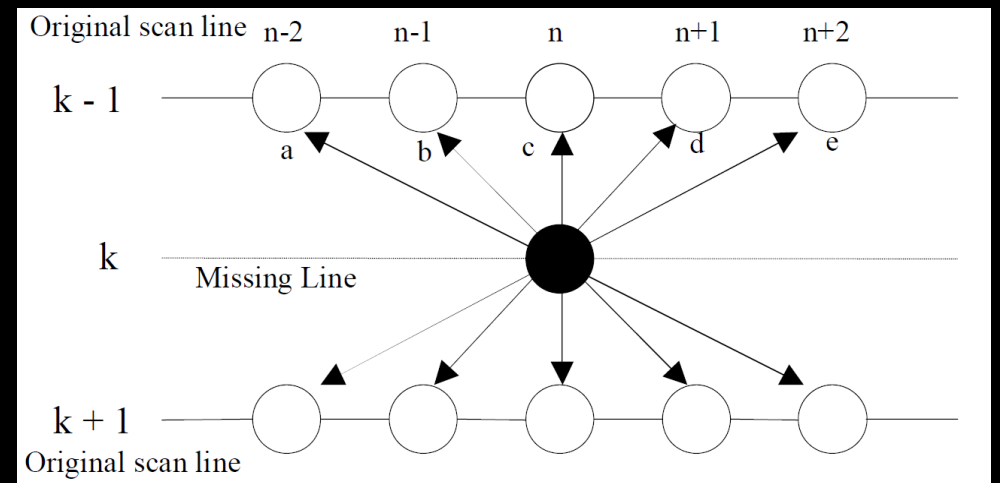


<https://www.whereig.com/>



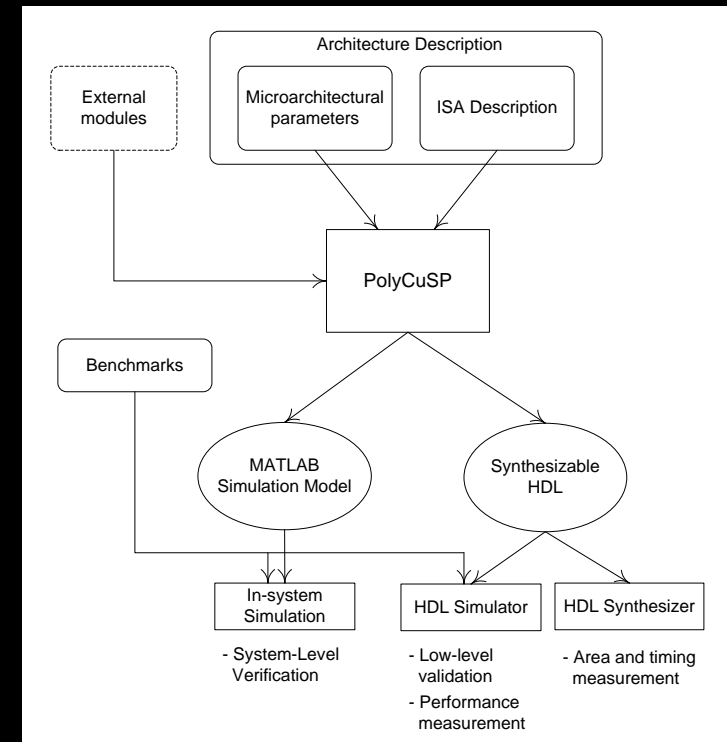
ASIP related research in Polytechnique

- **H. Mahvash Mohammadi**, Y. Savaria and J.M.P. Langlois, “A Hybrid Deinterlacing Algorithm Exploiting Reverse Motion Estimation,” *IET Image Processing*, May 2011, pp. 611-618.
- **H. Mahvash Mohammadi**, Y. Savaria and J.M.P. Langlois, “Enhanced Motion Compensated Deinterlacing Algorithm”, *IET Image Processing*, vol. 6, issue 8, November 2012, pp. 1041-1048
- **P. Aubertin**, J.M.P. Langlois and Yvon Savaria, “Real-Time Computation of Local Neighbourhood Functions in Application-Specific Instruction-Set Processors,” *IEEE Transactions on VLSI*, November 2012, pp. 2031-2043.
- + ICECE2006, NEWCAS 2006, IEEE TCE 2007, NEWCAS 2009

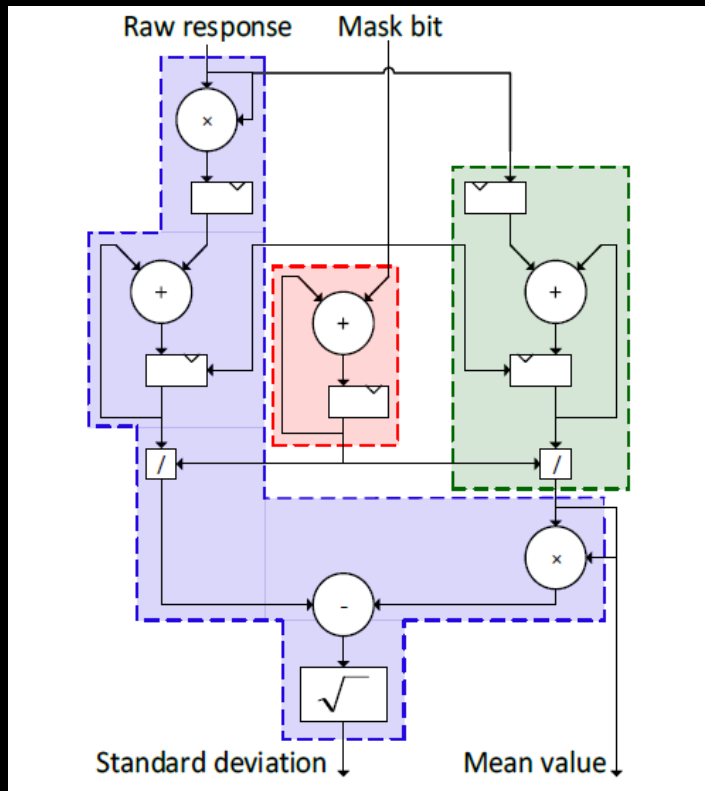


ASIP related research in Polytechnique

- **S. Vakili, J.M.P. Langlois and G. Bois, “Enhanced precision analysis for accuracy-aware bit-width optimization using affine arithmetic,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, December 2013.**
- **S. Vakili, J.M.P. Langlois and G. Bois, “Accuracy-aware processor customization for fixed-point applications”, *IET Computers & Digital Techniques*, January 2016.**



ASIP related research in Polytechnique



- **H. Bendaoudi**, F. Cheriet, H. Ben Tahar and J.M.P. Langlois, “A Scalable Hardware Architecture for Retinal Blood Vessel Detection in High Resolution Fundus Images,” *Proceedings of Conference on Design & Architectures for Signal & Image Processing*, Oct. 2014.
- **Bendaoudi**, Q. Gan, F. Cheriet, H. Ben Tahar and J.M.P. Langlois, “A Run-Length Encoding Co-Processor for Retinal Image Texture Analysis,” *IEEE International Conference on ReConfigurable Computing and FPGAs*, Dec. 2015.
- **H. Bendaoudi**, F. Cheriet, **A. Manraj**, H. Ben Tahar and J.M.P. Langlois, “Flexible architectures for retinal blood vessel segmentation in high resolution fundus images,” *Journal of Real Time Image Processing*, June 2018.

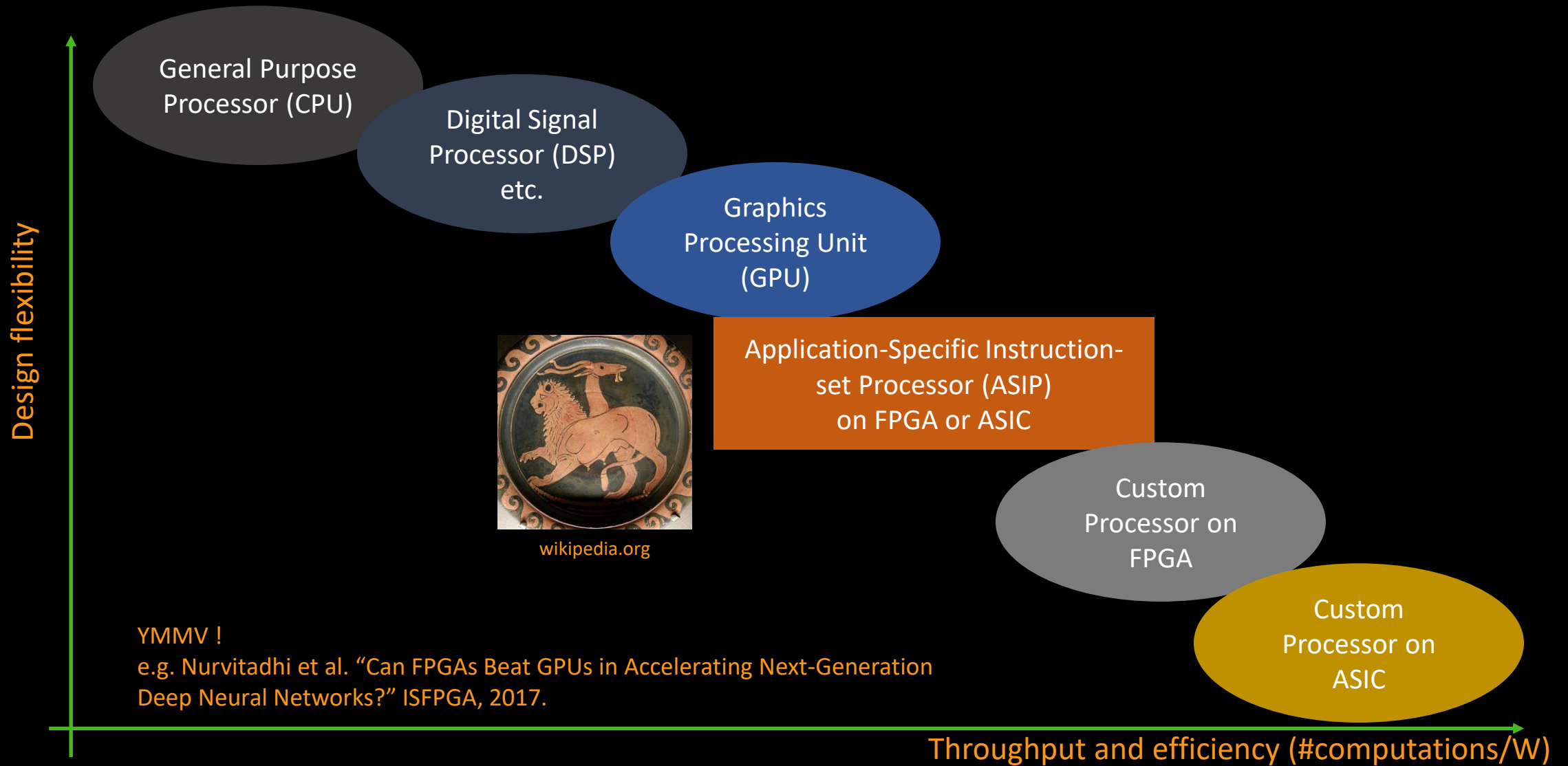


ASIPs are the ideal abstraction
to learn computer architecture

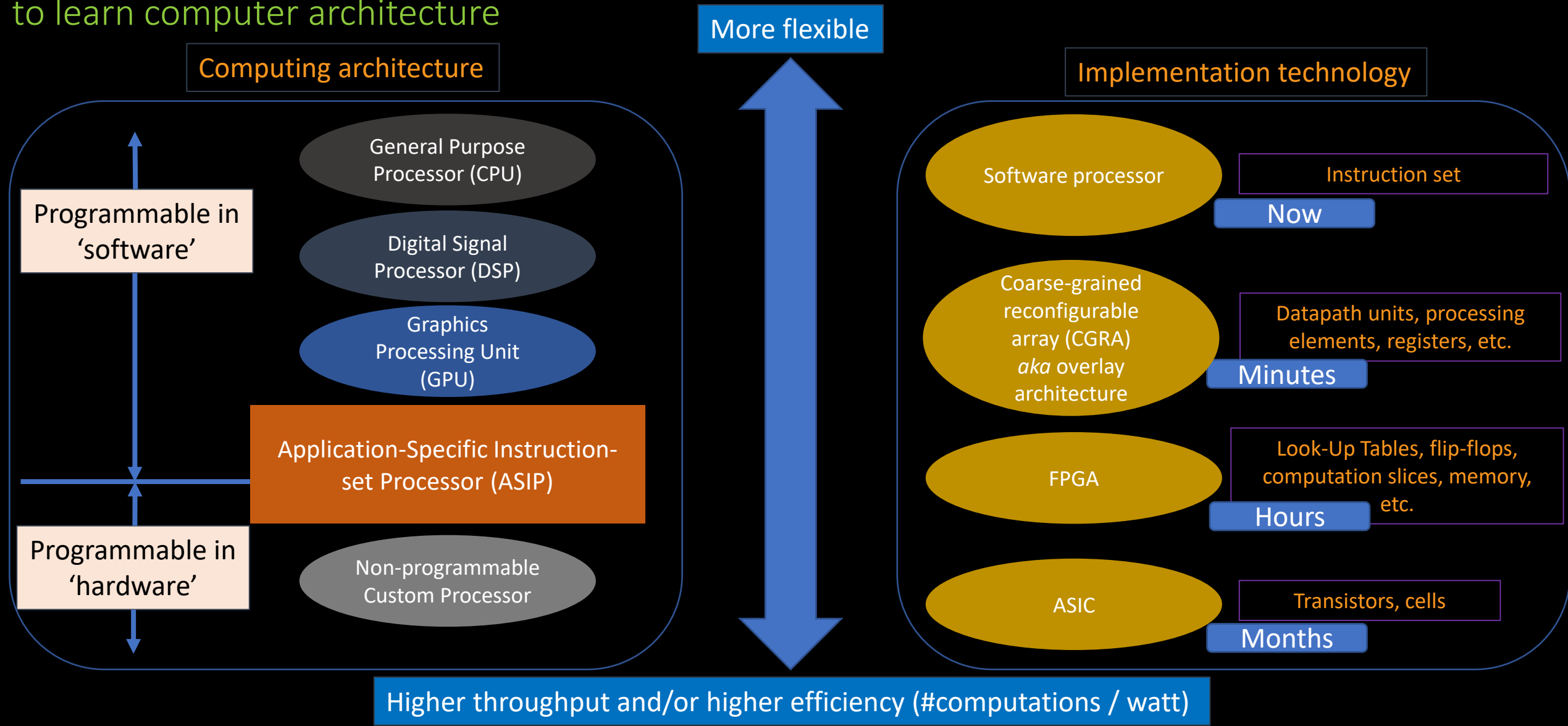
A few traditional computer architecture topics

- Functional units
- Memory hierarchy
- Performance metrics
- Pipelining and parallel processing
- Instruction set architecture
- Microarchitecture
- Compilers

ASIPs are the ideal abstraction to learn computer architecture



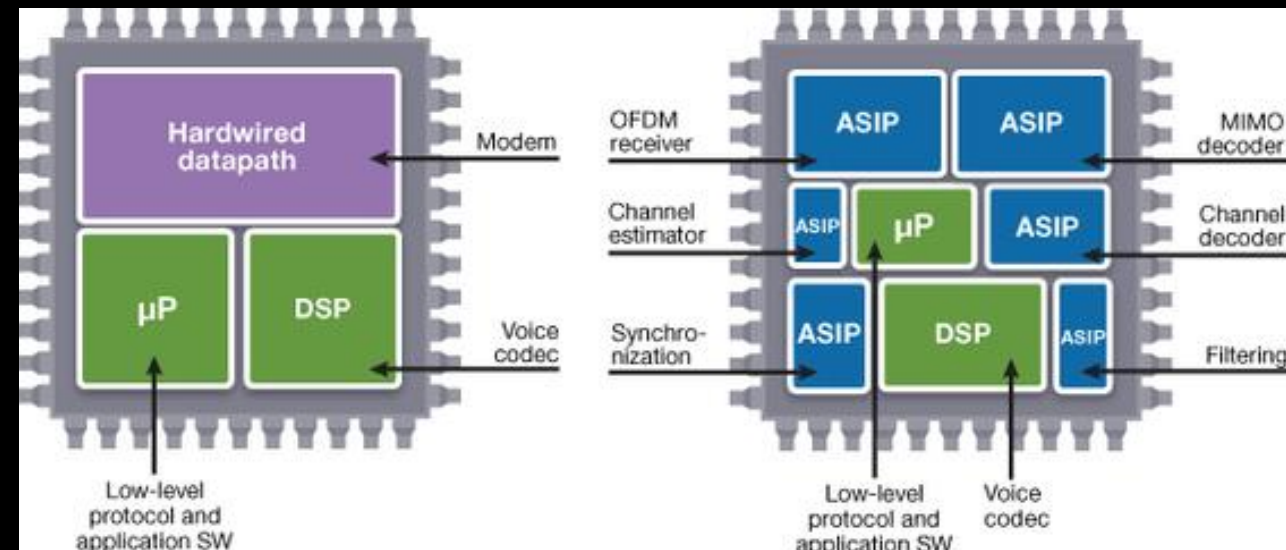
ASIPs are the ideal abstraction to learn computer architecture



ASIPs are the ideal abstraction to learn computer architecture

Heterogeneous embedded multiprocessor

- Keep one CPU for overall control
- Exploit explicit parallelism with multiple hardware accelerators and coprocessors
 - Special needs for custom processor : vector instructions, advanced pipelines, special register files and scratchpad memories, etc.
 - Wide data accesses
 - A CPU can be overkill
 - A custom processor can be painful to design
- => Use multiple ASIPs as custom processors



About the course

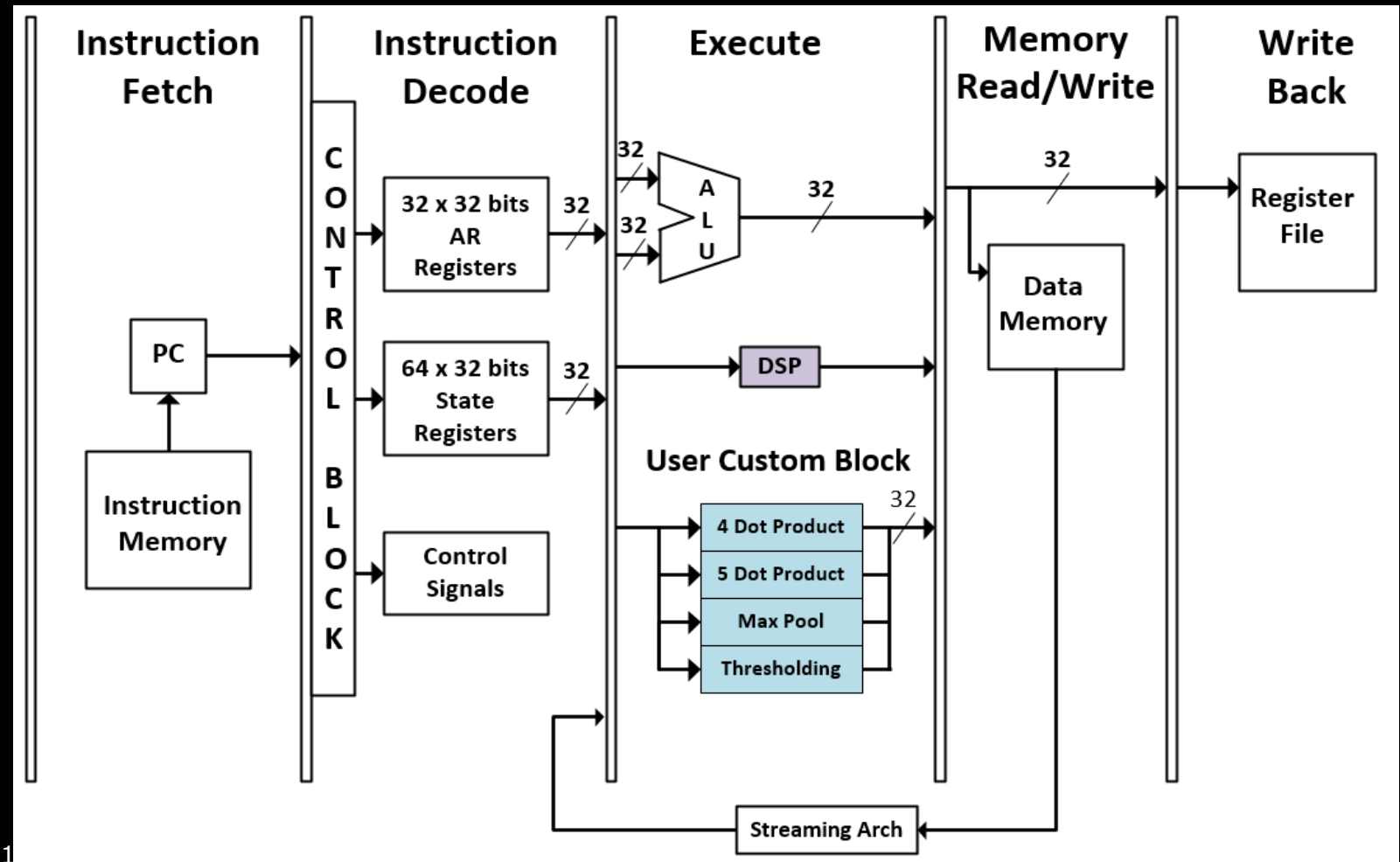
- INF8505 : Embedded Configurable Processors
An advanced computer architecture course
- Senior undergraduate (European M2) / graduate course
- Computer and Electrical Engineering students
- 20-30 students per year
- Taught mostly in French, but English-friendly
- Taught by me or my grad students
- 3-credit course = 135 hours of student work:
 - 33 hours of class discussions
 - 25 hours reading research papers and preparing reports
 - 18 in-lab + 9 out-of-lab hours for laboratory exercises
 - 40 hours for course project : design your ASIP
 - 10 hours personal study and final exam

- Microprocessor design
- ASIPs and configurable processors
- Microprocessor performance metrics
- High performance processors, superscalar, VLIW, etc.
- Architecture (Processor) Description Languages :
 - MIMOLA, LISA, nML
- ASIP applications : cryptography, image processing, general DSP, neural networks
- Loop analysis : how much can you really accelerate your application ?
- The compilation problem and retargetable processors
- Automated ASIP configuration

- 2008 – 2018 : Tensilica/Cadence tools
 - model based on a fixed processor core (Xtensa)
 - can configure the processor
 - can add custom instructions
 - ‘heavy’ processor generation
 - could not have RTL model = can’t get clock frequency
- Since Jan. 2019 : Synopsys’ ASIP Designer
 - many models available to start from
 - significant flexibility
 - can get all performance metrics

- Objectives :
 - Design and implement an ASIP for an application of the student's choice, obtain performance results and measure costs
 - Learn to use high-end tools (Synopsys' ASIP Designer, etc.)
 - Document and communicate the work in a 4-6 page research paper and make a 20-minute oral presentation
- Completed in teams of 2 or 3 students, in 40 hours over 13 weeks
- Students are encouraged to chose a topic related to their research, including:
 - Deep learning
 - Network processors
 - High performance computation (genomics, string matching, simulation, etc.)
 - Computer arithmetic (sparse matrices, transcendental functions)
 - DSP and image processing
 - Cryptography
- Deliverables every three weeks
- Students are encouraged to submit their paper to an international conference such as ICCD or DASIP : half a dozen accepted papers since 2008

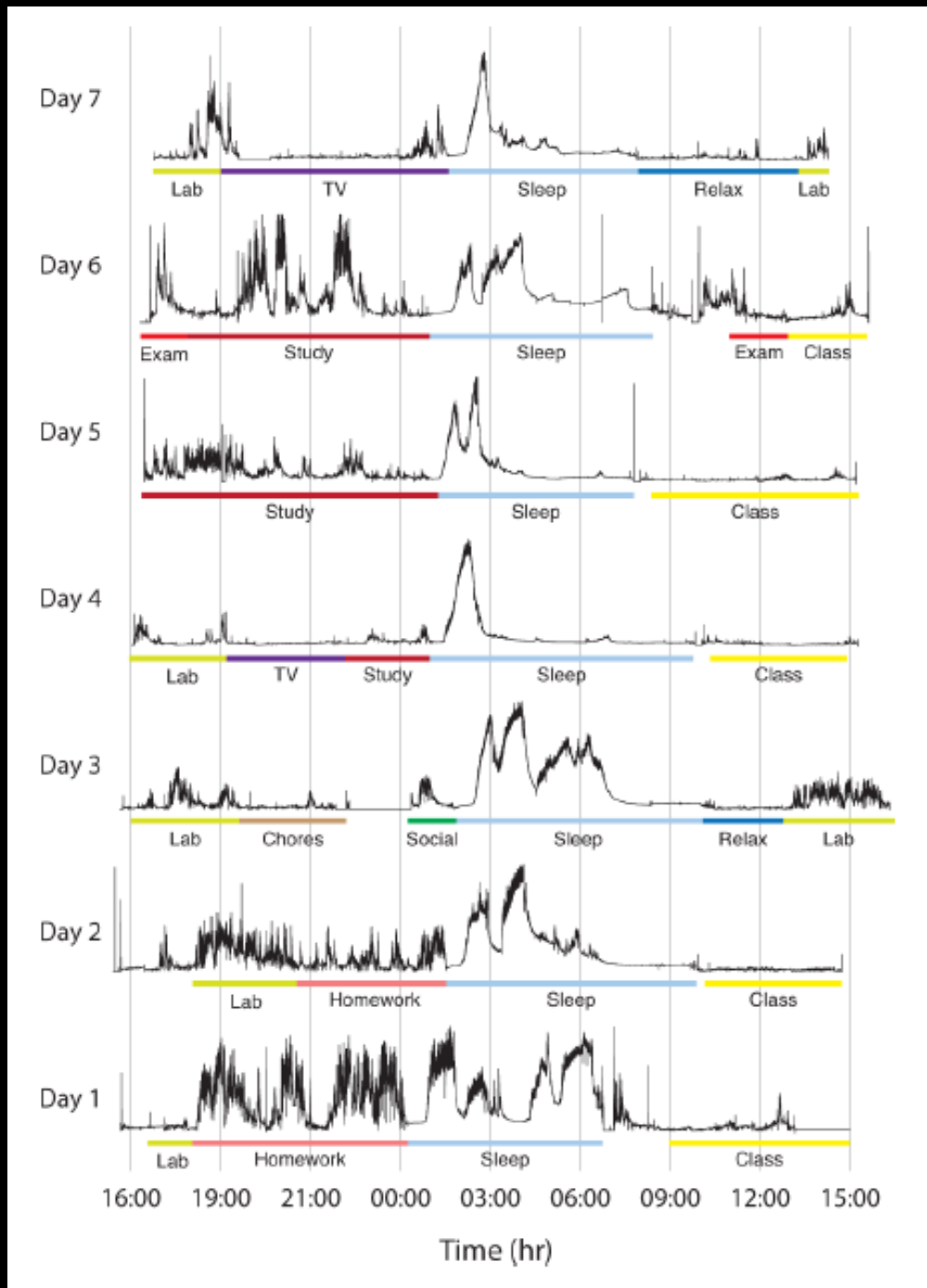
- **S. Chidambaram, A. Riviello, J.M.P. Langlois and J.-P. David**, "Accelerating the Inference Phase in Ternary Convolutional Neural Networks using Configurable Processors," *DASIP, 2018*.

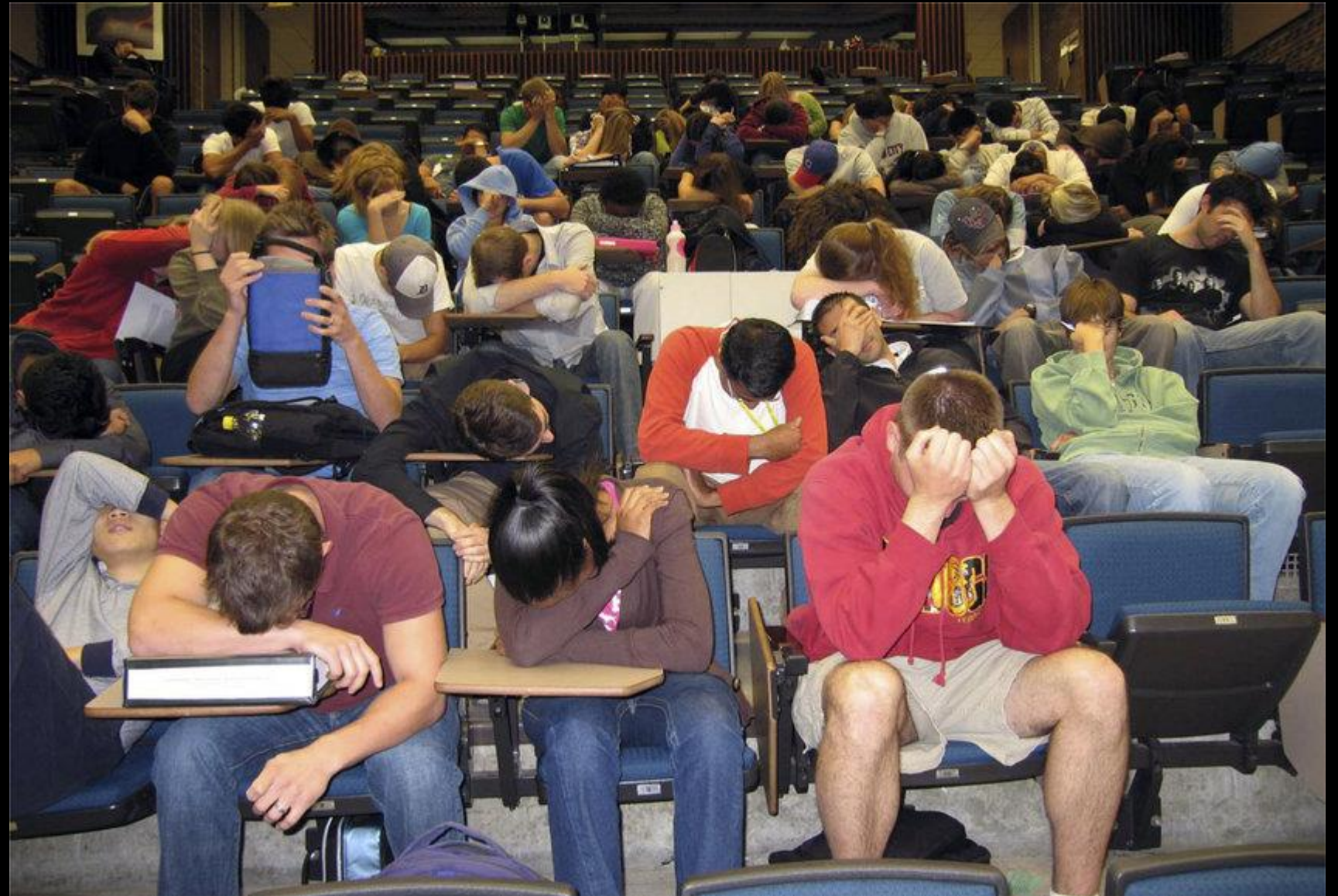


Flipped-classroom meets ASIP research



Ming-Zher Poh; Swenson, N.C.; Picard, R.W., "A Wearable Sensor for Unobtrusive, Long-Term Assessment of Electrodermal Activity," *IEEE Transactions on Biomedical Engineering*, May 2010





Flipped-classroom meets ASIP research

Read, comment and discuss one paper each week

- Students must read one (or portions of one) paper and hand-in a one-page report every week
- The paper is then discussed in class and serves as a basis to cover that week's subject matter
- Counts for 20% of final mark
- Sample papers :
 - S. Radhakrishnan et al. "Customization of application specific heterogeneous multi-pipeline processors," DATE 2006.
 - Q. Jinguo et al. "Fine-grained analysis and design of ASIP instruction set for application of encryption," ICNC 2012.
 - A. Gupta et al. "Accelerating SVM on Ultra Low Power ASIP for High Throughput Streaming Applications," in IEEE VLSI 2015.
 - Y. Xin et al. "An Application Specific Instruction Set Processor (ASIP) for Adaptive Filters in Neural Prosthetics," in IEEE/ACM Transactions on Computational Biology and Bioinformatics 2015.
 - I. Latifis et al., "Matlab to C compilation targeting Application Specific Instruction Set Processors," DATE 2016.
 - S. Chidambaram et al. : "Accelerating the Inference Phase in Ternary Convolutional Neural Networks Using Configurable Processors," DASIP 2018.
 - J. Hu et al., "An application specific instruction set processor (ASIP) for the niederreiter cryptosystem," ISDFS 2018.

Title				
Authors and affiliation				
Publication and category				
Impact - citations				
Contents	Theory : Marketing :	Algorithms : Methodology:	Architecture : Others :	Implementation :
Subject / domain				
Problems considered	<ul style="list-style-type: none"> • One • Two • Three 			
Proposals	<ul style="list-style-type: none"> • One • Two • Three 			
Results	<ul style="list-style-type: none"> • One Two 			
Personal appreciation	<ul style="list-style-type: none"> • Your opinion here 			

Lessons learned

- Remember Amdahl's law
- The chosen algorithm should not be too complex – should be clearly representable with a half-page datapath
- The application must benefit from acceleration :
 - High throughput more than low latency
- There should be a balance between computation effort and data access needs
- Avoid algorithms requiring lots of divisions and transcendental functions (unless that is the goal of the project)
- Avoid floating point computation (unless that is the goal of the project)

- Students are very satisfied with their learning outcomes
 - Enjoy reading, commenting and discussing research papers
 - Enjoy performing the project, although it is a lot of work
 - Enjoy practicing the regular research steps
 - Literature : what is the problem
 - Objectives : what we want to about the problem
 - Proposals : what are our ideas
 - Results : what we achieved
 - Compare : how did we do

Definition [\[edit\]](#)

Amdahl's law can be formulated in the following way:^[2]

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

where

- S_{latency} is the theoretical speedup of the execution of the whole task;
- s is the speedup of the part of the task that benefits from improved system resources;
- p is the proportion of execution time that the part benefiting from improved resources originally occupied.

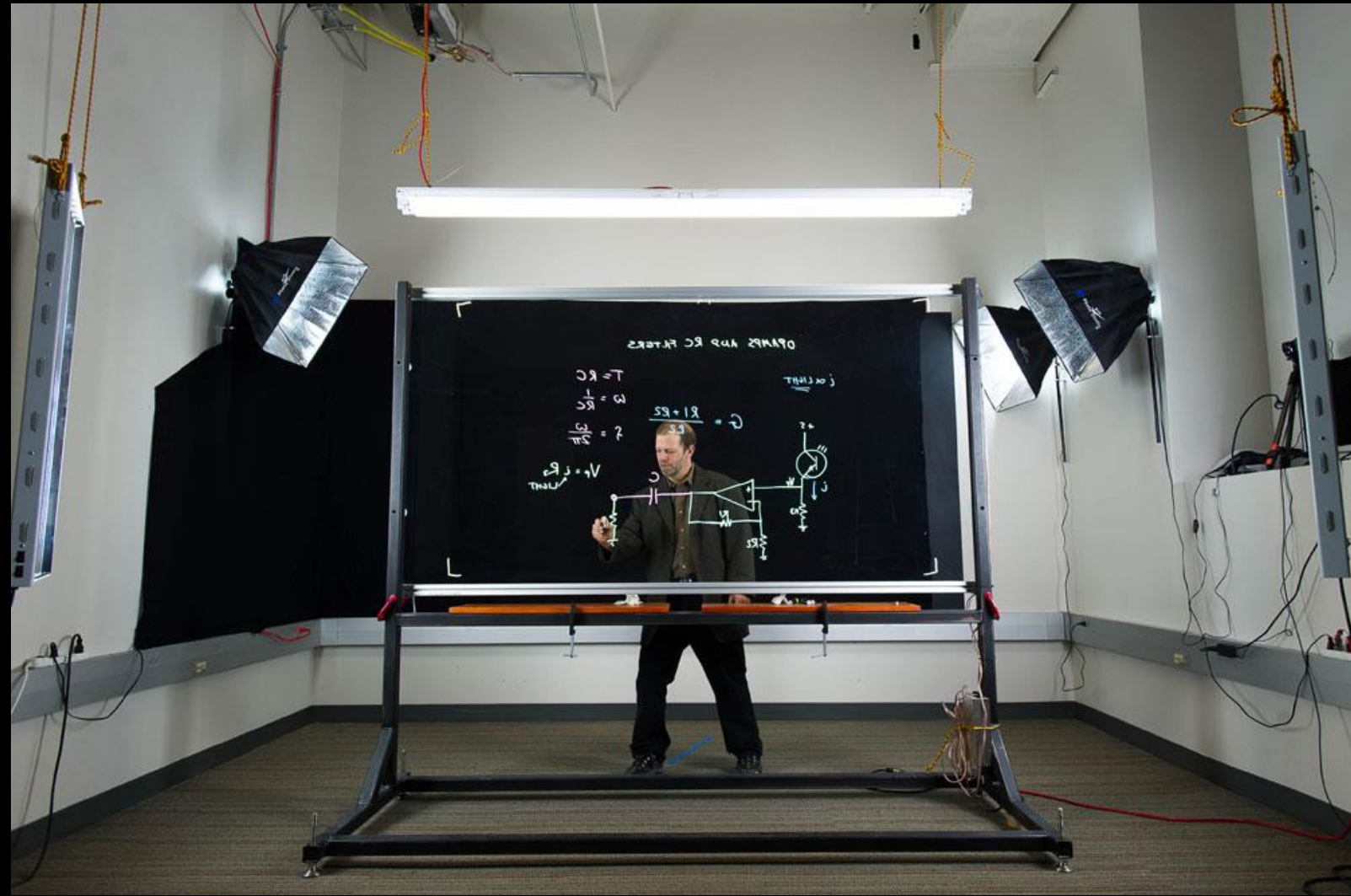
Furthermore,

$$\left\{ \begin{array}{l} S_{\text{latency}}(s) \leq \frac{1}{1-p} \\ \lim_{s \rightarrow \infty} S_{\text{latency}}(s) = \frac{1}{1-p} \end{array} \right.$$

[wikipedia.org](https://en.wikipedia.org/wiki/Amdahl%27s_law)

- Remember Amdahl's law
- The chosen algorithm should not be too complex – should be clearly representable with a half-page datapath
- The application must benefit from acceleration :
 - High throughput more than low latency
- There should be a balance between computation effort and data access needs
- Avoid algorithms requiring lots of divisions and transcendental functions (unless that is the goal of the project)
- Avoid floating point computation (unless that is the goal of the project)
- Students are very satisfied with their learning outcomes
 - Enjoy reading, commenting and discussing research papers
 - Enjoy performing the project, although it is a lot of work
 - Enjoy practicing the regular research steps
 - Literature : what is the problem
 - Objectives : what we want to about the problem
 - Proposals : what are our ideas
 - Results : what we achieved
 - Compare : how did we do

About the lightboard



© lightboard.info
prof. Michael Peshkin, NWU, Illinois USA

About the lightboard



Outline

- About me
- ASIP related research in Polytechnique Montréal
- ASIPs are the ideal abstraction to learn computer architecture
- INF8505: an ASIP design course
- A course project with Synopsys' ASIP Designer
- Flipped-classroom meets ASIP research
- Lessons learned
- About the lightboard

Learning Computer Architecture through the ASIP Paradigm: A Research-oriented Approach

Synopsys ASIP University Day
2021-11-17

Pierre Langlois, Eng., Ph.D.
Professor and Chair
Department of Computer and Software Engineering



**POLYTECHNIQUE
MONTRÉAL**

Presentation abstract

- The course INF8505 – Embedded Configurable Processors – is a three-credit, 135-hour graduate computer architecture course that has been running in Polytechnique Montréal since 2008. It is also available as an elective to senior Electrical and Computer Engineering undergraduates. The course focuses on Application Specific Instruction set Processor (ASIP) design, and its main topics are custom datapath and custom memory hierarchy design, processor description languages, retargetable compilers, and processor performance metrics. It exploits high throughput applications such as deep learning, image processing and cryptography to demonstrate the ASIP's potential. From its beginning, the course has been taught in a flipped-classroom style where students are assigned one research paper every week for which they must produce a one-page report. The paper is then discussed in class and the instructor weaves the course topics with the paper's main contributions. A major 52-hour course project is anchored in laboratory exercises. Two-student teams use Synopsys' ASIP Designer to design and simulate an ASIP tailored to an application of their choice, then submit a project report as a 4- or 6-page research paper. To date, there are almost 250 course alumni and half a dozen project report papers have been presented in international conferences.