LUND UNIVERSITY

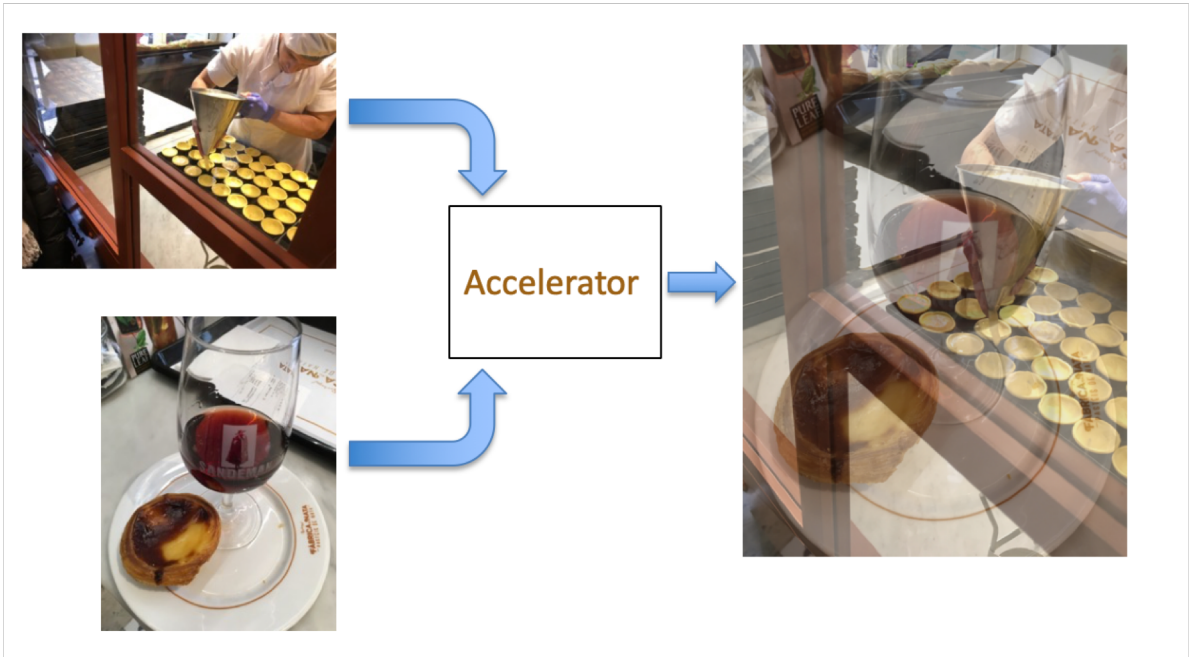
# IC Project- ETIN35

## PROJECTS, JOACHIM RODRIGUES

# Overview

- Overlay Accelerator
- Edge detection
- RISC-V
- Pipelined FFT Processor

# Overlay Accelerator




- VGA resolution, RGBA 8bit
- 60fps
- Adjustable transparency

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sample Length: | 8 | | | | | | | | 8 | | | | | | | | 8 | | | | | | | | 8 | | | | | | | |
| Channel Membership: | Red | | | | | | | | Green | | | | | | | | Blue | | | | | | | | Alpha | | | | | | | |
| Bit Number: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

LUND UNIVERSITY

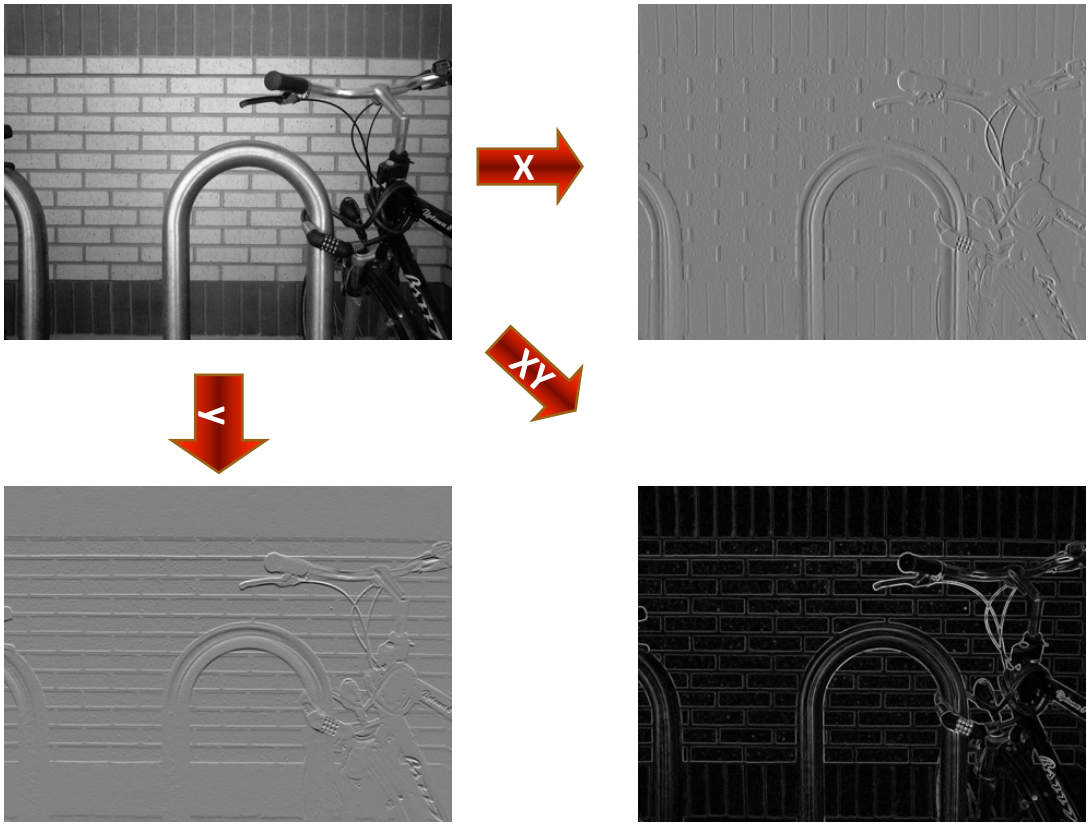
# Overlay Accelerator

## Tasks

- Develop an ASMD-chart specification of your design.
- HW-accelerator that simply uses a red and a green frame with 50% transparency the pixels in the image turning white into black and black into white
- Try different input frames

- Implement in VHDL and calculate the throughput, target will be 60fps

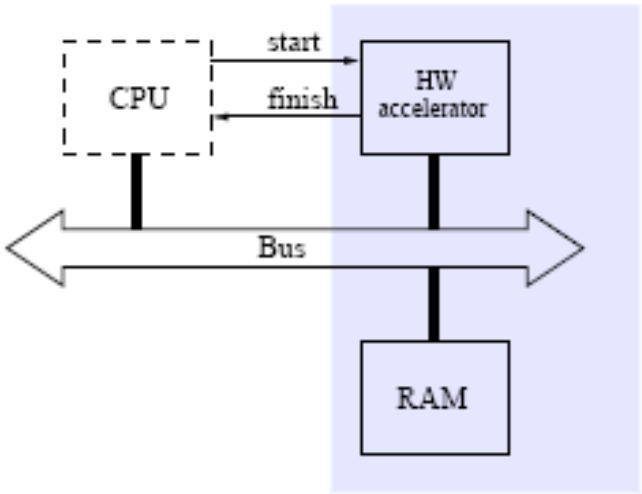
# Edge detection




LUND UNIVERSITY

# System Architecture

- Interface towards CPU will be ”realized” by push button and LED.
- HW accelerator is a slave that raises *finish.*
- Orignal and processed image will be stored at different locations in a RAM.




LUND
UNIVERSITY

# Theory

Edges are detected by

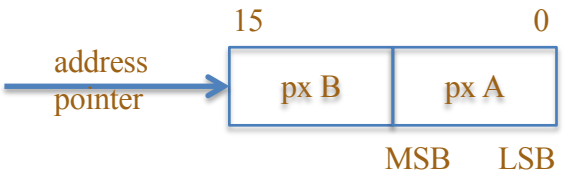
- convolution of a fixed coefficient 3-by-3 matrix over the pixel to compute x- and y-derivatives.
- Adding the maxima/minima of the derivatives forms a new ”pixel”.

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

LUND UNIVERSITY

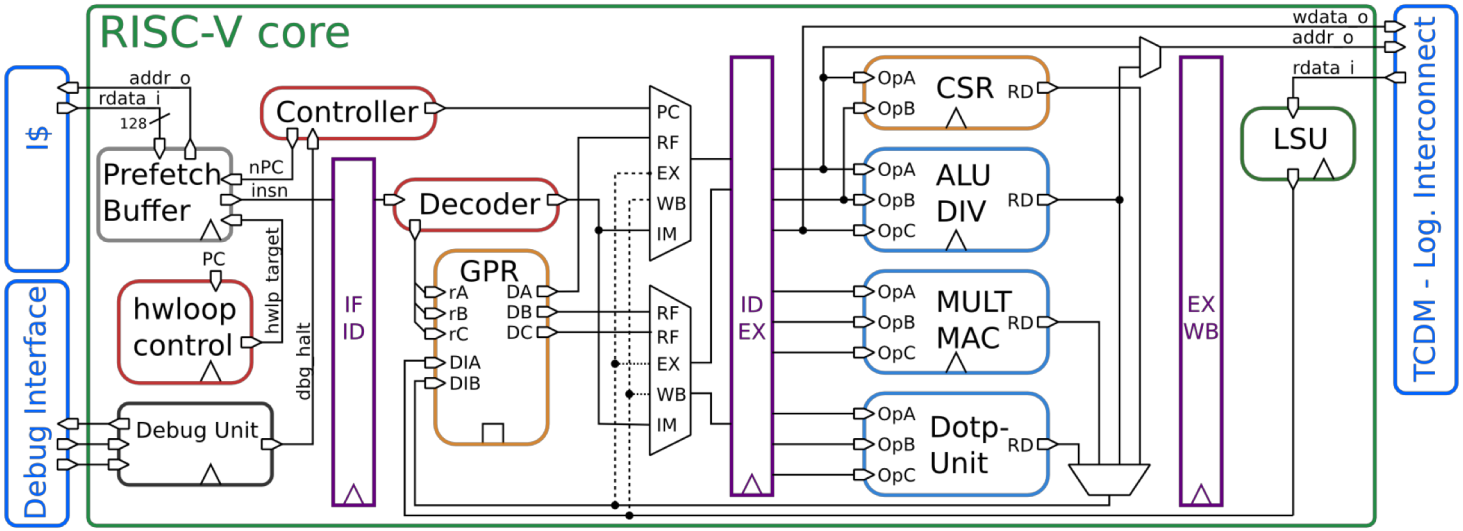
# Requirements

- Monochrome CIF (352x288), with a maximum number of frames per second (fps).
- Intensity of a pixel is represented by 8-bit.
- The pixels are stored successively in a 16-bit RAM.
- Each *read*- or *write* addresses 2 pixels, i.e., two pixels per 16-bit word.
- The memory needs to be allocated such that original and and processed pixels may be accommodated in two locations.
- The HW-accelerator is triggered by a signal *start* from the CPU and raises *finish* as soon as the last pixel in a frame is processed.




LUND UNIVERSITY

# RISC V PULPino




Main Features:

- Micro-controller style platform (minimal system);
- 32-bit instructions;
- 4-stage pipeline;
- Easier extension;

LUND UNIVERSITY

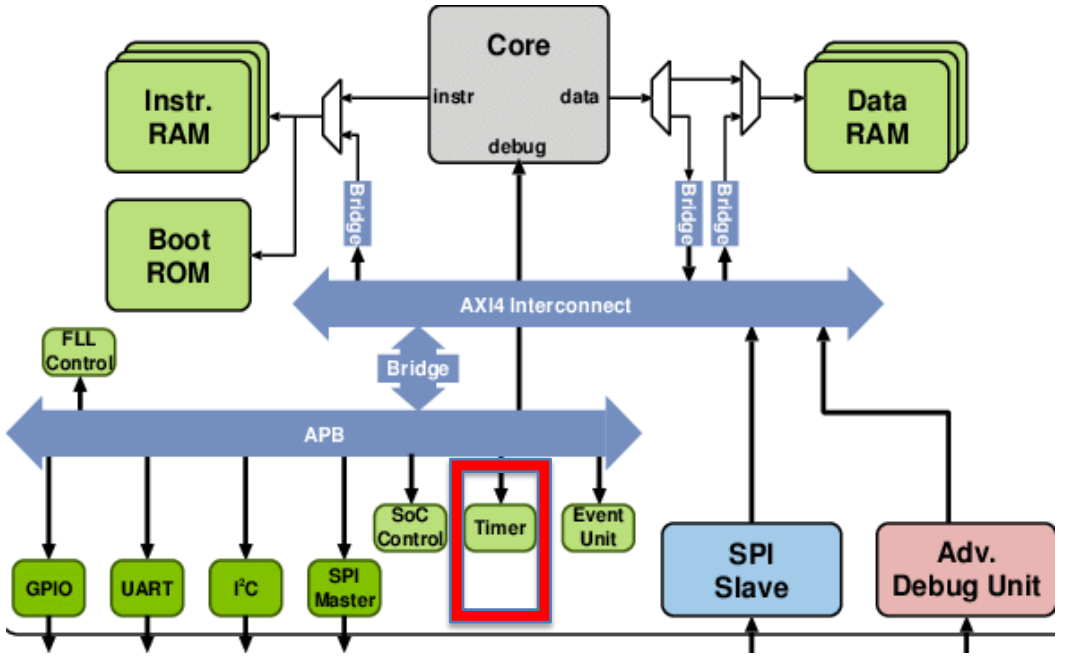
# RISCV - Projects

**Tasks:**

1. Replace the Timer component for a dummy component (e.g byte-swap) to the PULPino, and verify behavior up to post-layout simulations (passing).

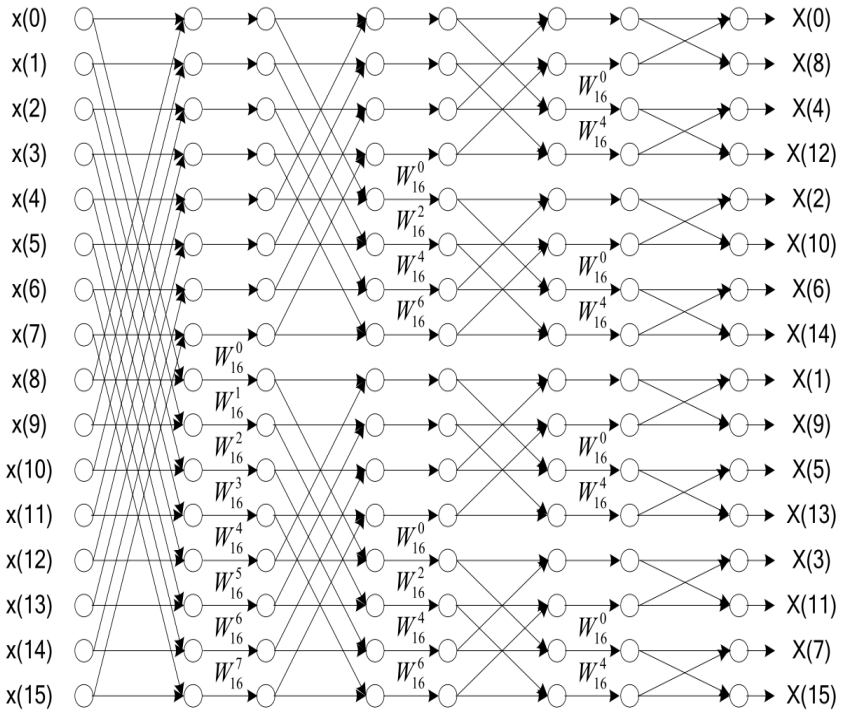
2. Replace the “dummy” component to the matrix multiplier from the compulsory part, but now performing a general matrix multiplication (5x5 by 5x5) for grade 4. Results should be compared to a functional model.

3. Program the matrix multiplier to perform a 3D 5x5 convolution for grade 5. No optimization requirements.



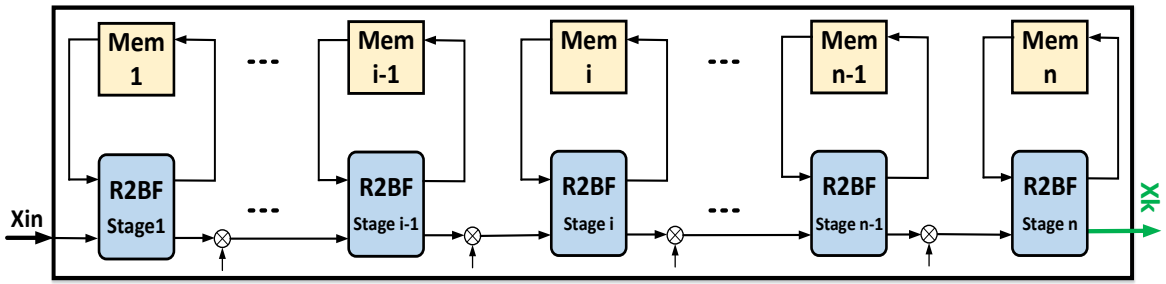

LUND UNIVERSITY

# Pipelined FFT Processor




$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \qquad k = 0, 1, 2, ..., N-1$$

$$W_N^{nk} = e^{-j2\pi nk/N} = \cos(2\pi nk/N) - j\sin(2\pi nk/N)$$




LUND UNIVERSITY

# Pipelined FFT Architecture- Features

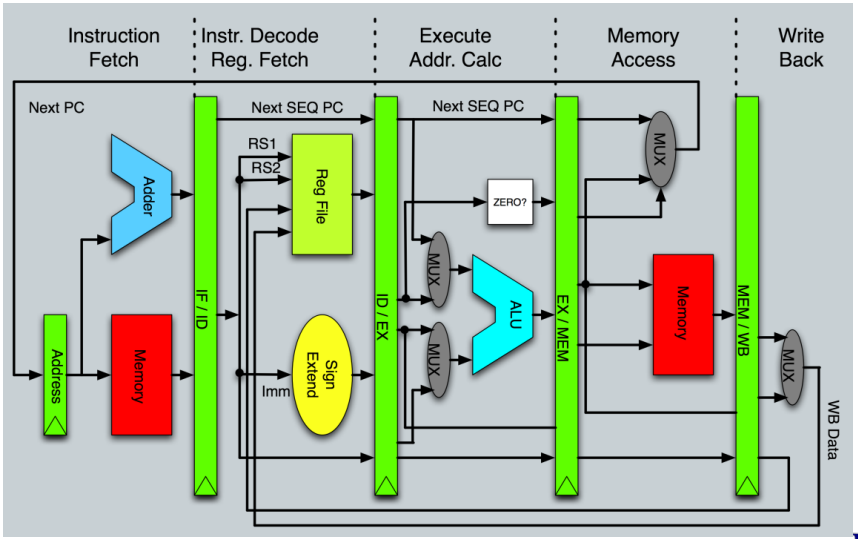
- Architecture: SDF (shown in figure)
- Algorithm: Radix 2
- FFT Size: 2048-point
- Input/Output samples are complex numbers
  - Fixed-point representation: 12 bits for real/imaginary parts of input/outputs
- Fixed-point model of FFT with corresponding simulation results in MATLAB are needed to verify the functionality

# Mini-MIPS project (Steffen Malkowsky)

- 32-bit RISC with a subset of MIPS instructions.
- Grading:
  - Grade 3: Fully verified pipelined Mini-MIPS.
  - Alt 1
    - Grade 4 : Xilinx Ethernet I/O
    - Grade 5 : Extended instruction set for full C compliance
  - Alt 2
    - Grade 4/5 : open for suggestions (implement accelerator)
  - Prerequisite course:
    - EITF35 Introduction to Structured VLSI Design
    - EITF20 Computer Architecture

# Mandatory tasks

- Task 1: behavior modeling
- Task 2: synthesizable pipelined implementation
- Task 3: P&R in 65nm CMOS
- Task 4: Verification in FPGA




LUND UNIVERSITY