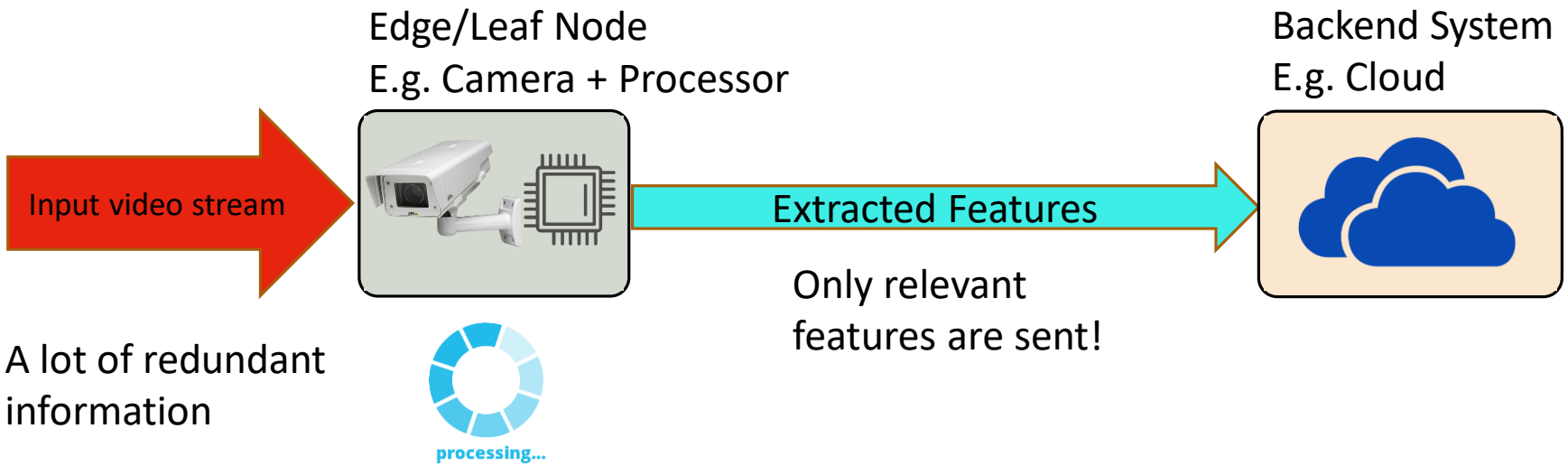
VLIW BASED RUNTIME RECONFIGURABLE MACHINE VISION COPROCESSOR ARCHITECTURE FOR EDGE COMPUTING

Dilshan Kumarathunga, Omega Gamage, Asitha Samarasinghe, Nipuna Saranga, Ranga Rodrigo, Ajith Pasqual

Department of Electronic and Telecommunication Engineering
University of Moratuwa
Sri Lanka

ASAP 2019, The 30th IEEE International Conference on
Application-specific Systems, Architectures and Processors

Introduction



Motivation

Challenges for edge computing architecture

- Support multiple vision algorithms at the same hardware platform
- Runtime reconfigurability
- Support high bandwidth video streams
- Extensibility
- Limited resources (area and power)

Approach

VLIW

- Instruction level parallelism
- Less hardware cost
- Simple hardware for instruction decoder

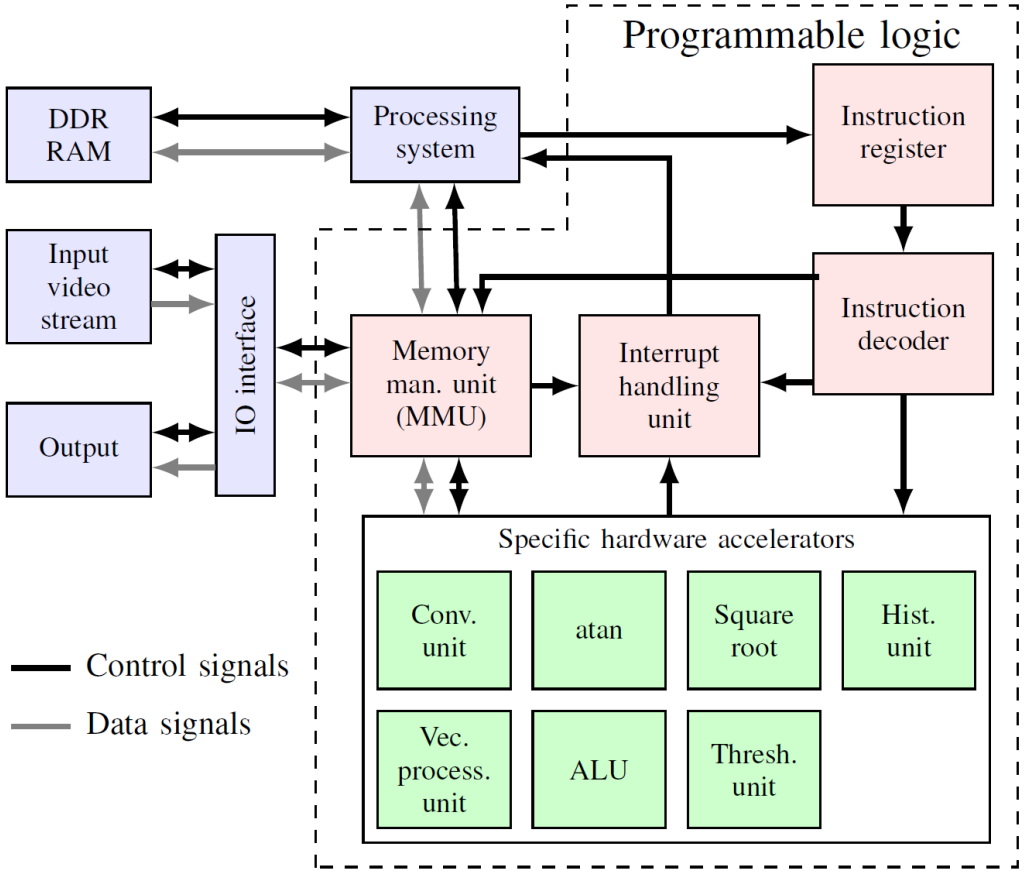
Mux Based Interconnection System

- Parallel data transactions among units
- One-to-many data transactions at the same time
- Low latency
- Scalability

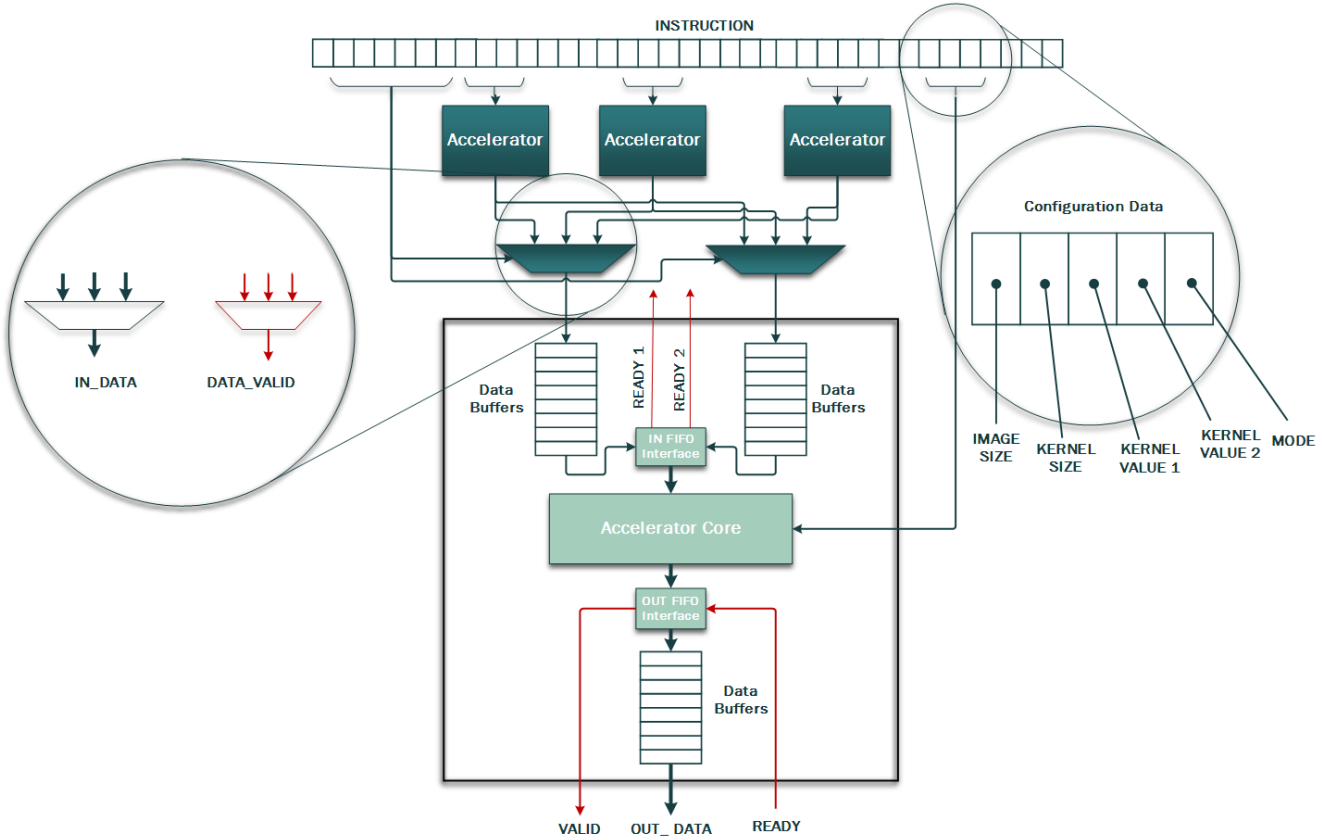
FPGA

- Support for extensibility
- Can do hardware upgrade via network(upload new bitstream after adding more units)
- Power efficient than GPU based solution

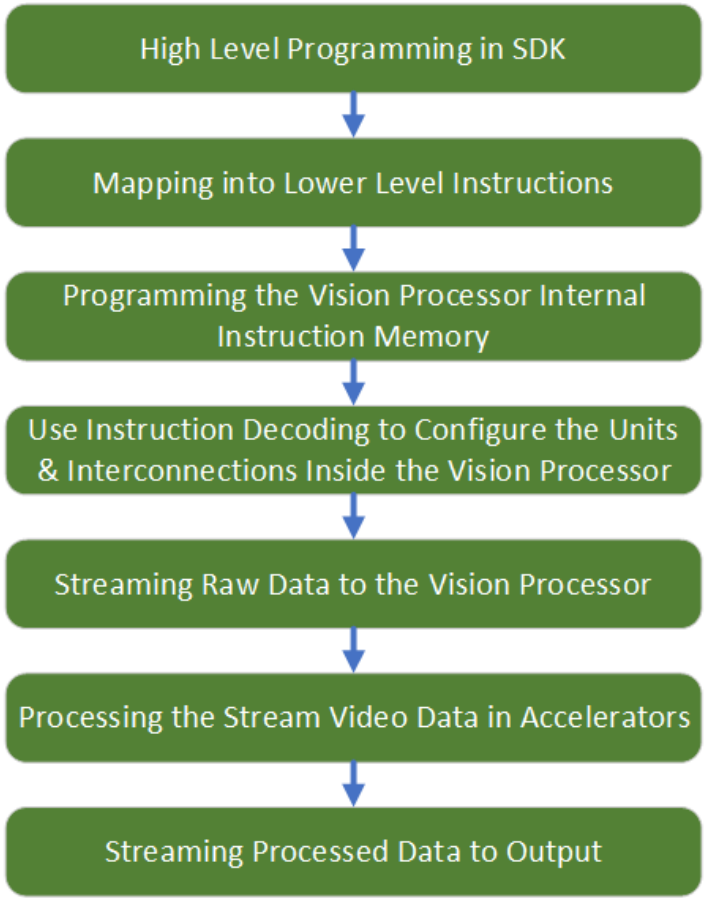
Overall Architecture



Internal Architecture



Software Interface & Programming



vision_lib.cpp

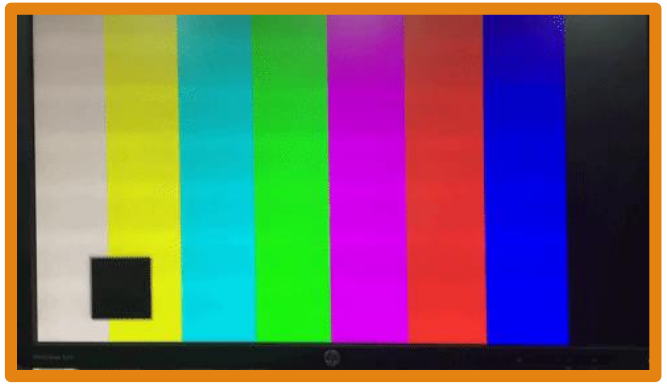
```
int VP_HOG(unsigned int* input_image, int image_width, int image_height,
int kernel_size, int* dx_kernel, int* dy_kernel, int number_of_bins, int* bin_limits)
{
int x = VPwrite(input_image);
int a = VPconvolution(1,x,kernel_size,image_width,image_height,dx_kernel,0);
int b = VPconvolution(2,x,kernel_size,image_width,image_height,dy_kernel,0);
int c = VPalu(1,a,a,2,0);
int d = VPalu(2,b,b,2,0);
int e = VPalu(3,c,d,0,0);
int f = VPsquareRoot(e);
int g = VPatan(b,a);
int h = VPhistogram(g,f,1,number_of_bins,bin_limits,0,0,image_height*image_width);

return h;
}
```

Results

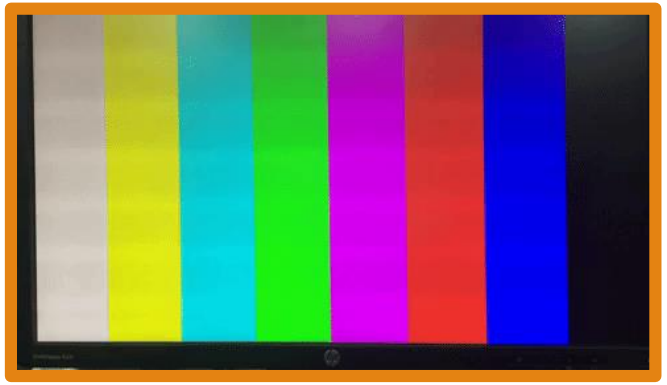
Metric	Chen[2]	Sun[3]	Xiao[5]	Ramirez-Martinez[6]	Reichenbach[7]	Proposed
Precision	8 bit	16 bit	32 bit	32 bit	16 bit	**32 bit**
Clock(MHz)	66.7	100	250	100	138.75	**148.5**
Max Res.(pixels)	640 X 480	1920 X 1080	640 X 480	640 X 480	1920 X 1080	**1920 X 1080**
Max fps	50	42	32	51	36.52	**60**
Runtime configurability	Fixed architecture	Fixed architecture	configurable	Fixed architecture	configurable	**Runtime reconfigurable**
Extendability	No	No	No	No	Yes	**Yes**

Results

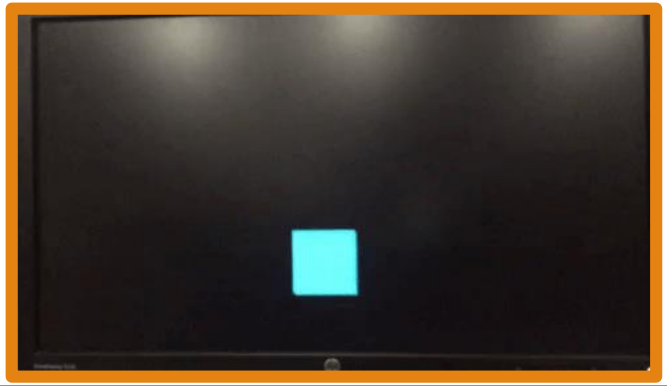
- Video with moving object



- Static background

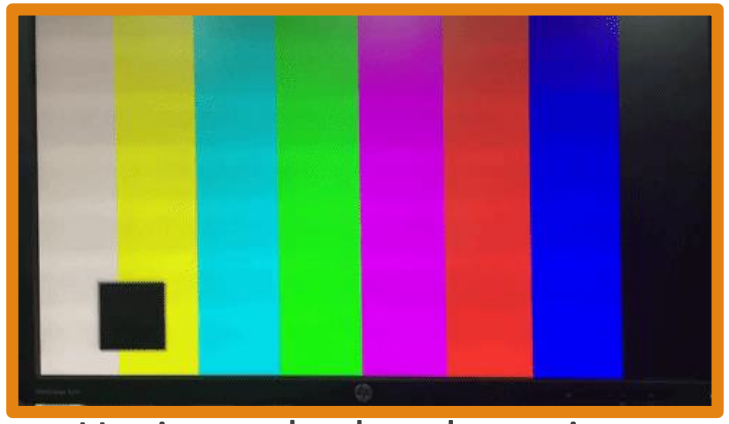


- After background subtraction

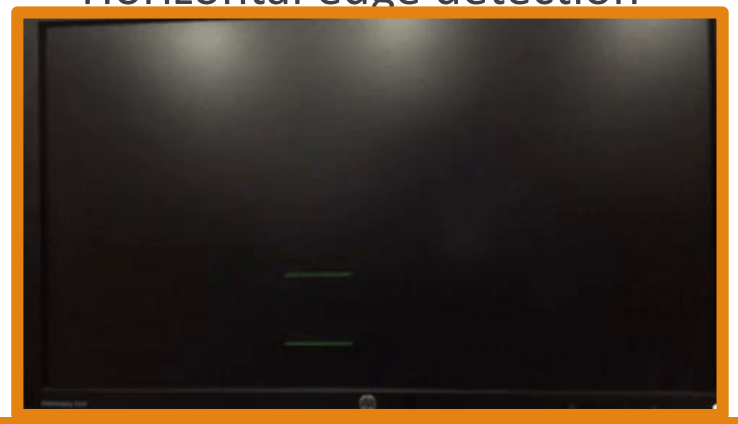


Results

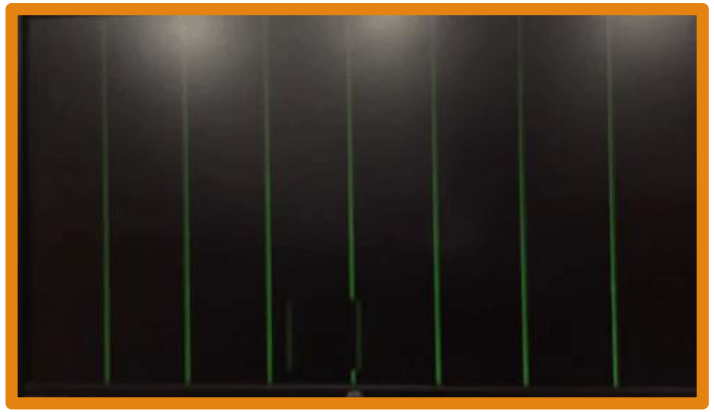
- Video with moving objects



- Horizontal edge detection



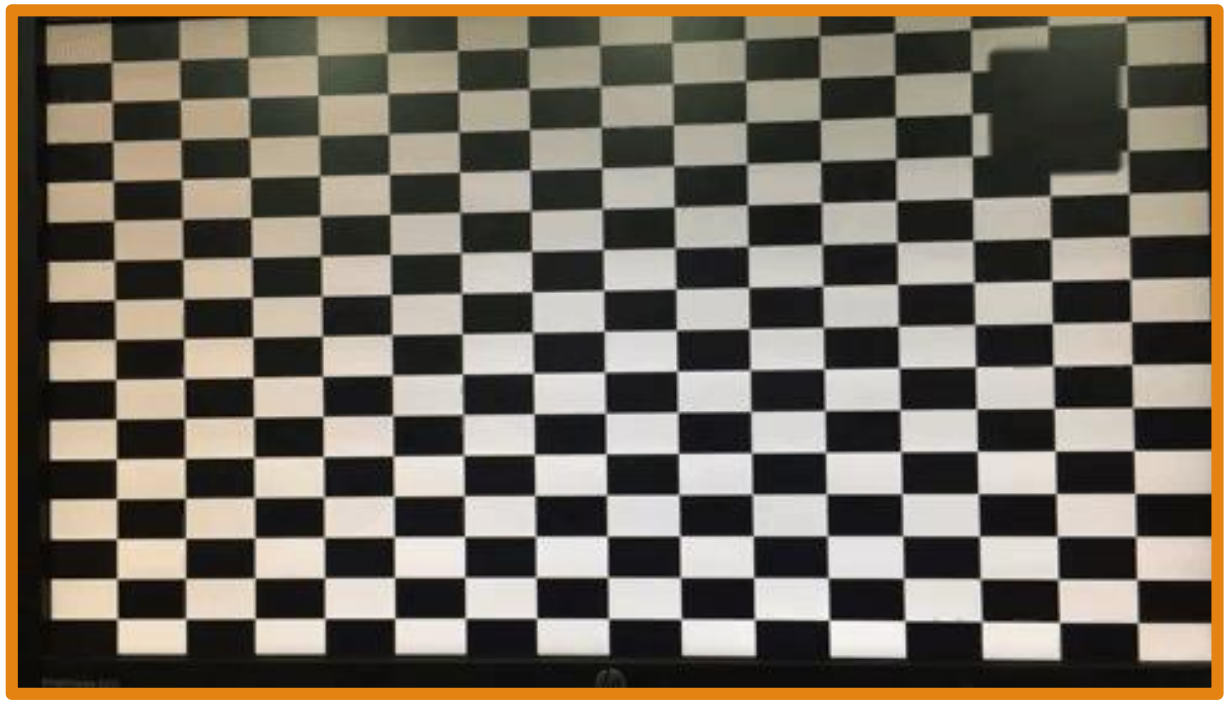
- Vertical edge detection



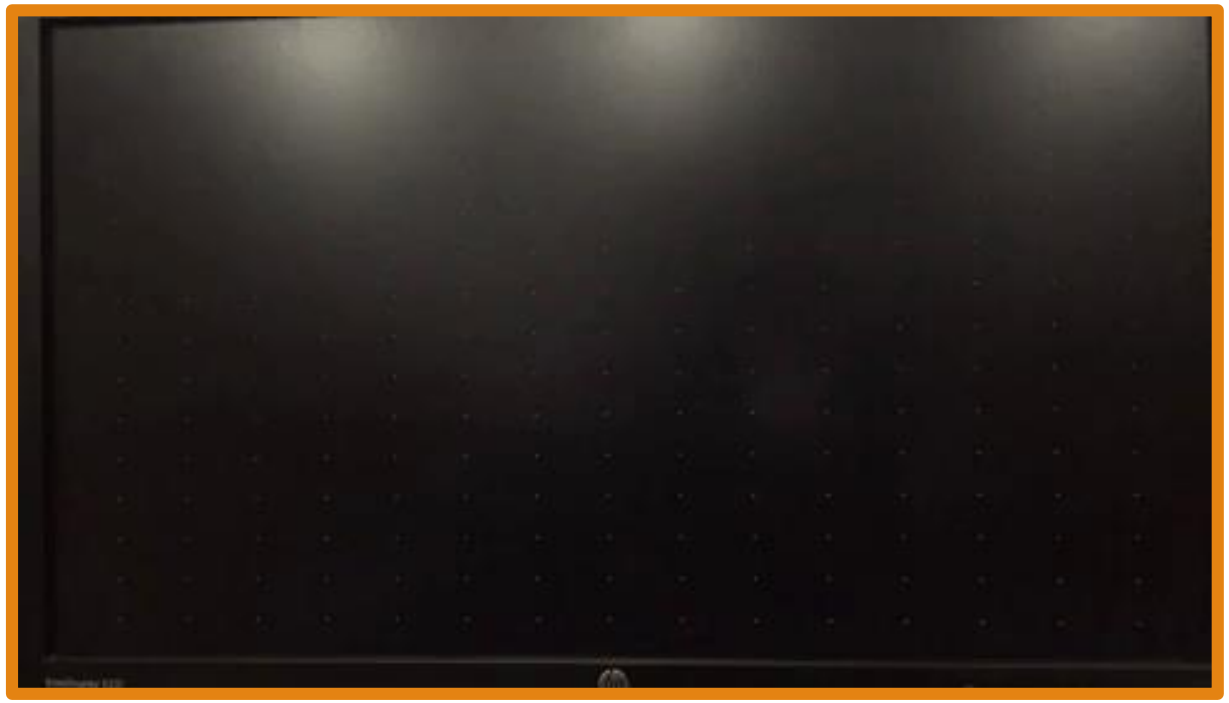
- Vertical and Horizontal edge detection

Results

- Original video



- Corner Detection



Results

Resolution	Software(µs)	Hardware(µs)	Gain
640 x 480 (VGA)	50,610	8,513	5.95
720 x 480 (WVGA)	64,918	10,730	6.05
1280 x 720 (HD)	118,106	19,274	6.13
1920 x 1080 (full HD)	268,953	43,493	6.18

- HoG feature extraction performance comparison with the software model

Results

Resource	Used	Available	Utilization
LUT	19,326	53,200	36.33%
LUTRAM	64	17,400	0.37%
FF	18,215	106,400	17.12%
Block RAM	37	140	26.43%
DSP	78	220	35.45%
BUFG	1	32	3.13%

- FPGA Resource Utilization In The ZedBoard

Conclusions

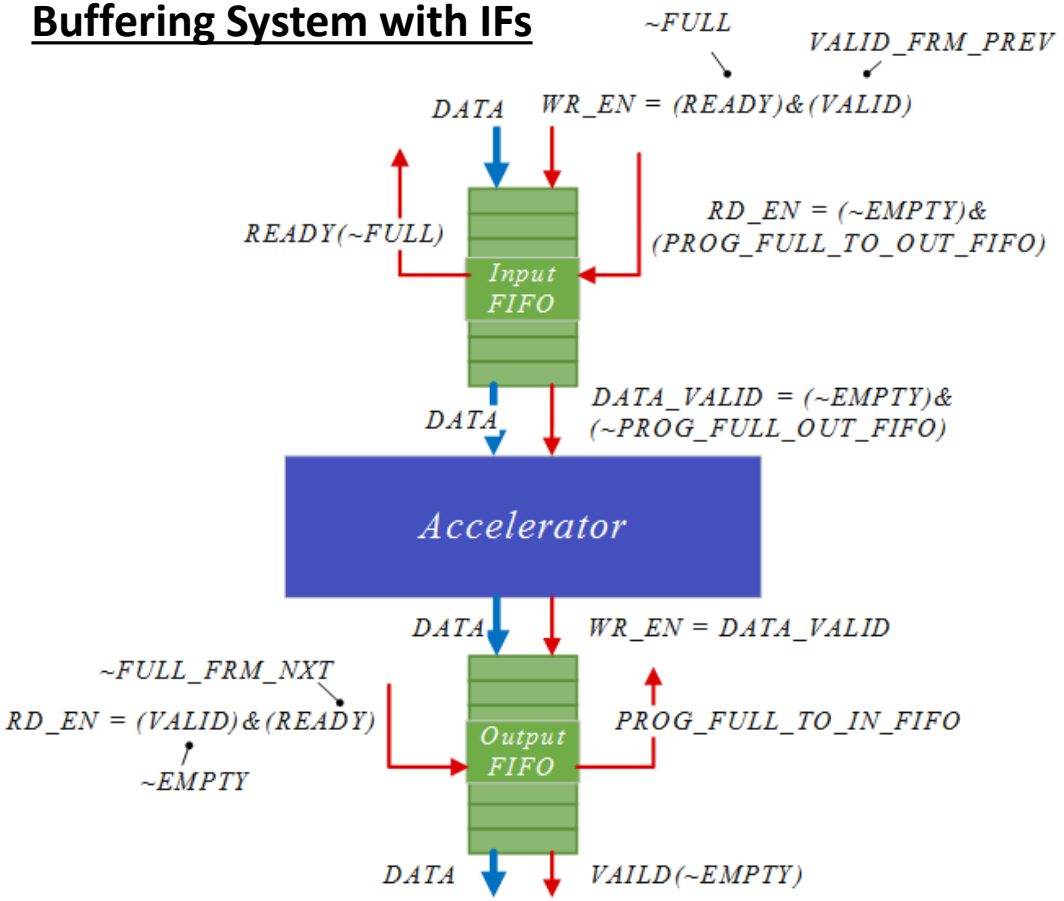
- Supports Full HD, 60fps video streams
- Support runtime reconfigurability
- Architecture is flexible to customize
- Architecture is extensible

THANK YOU!

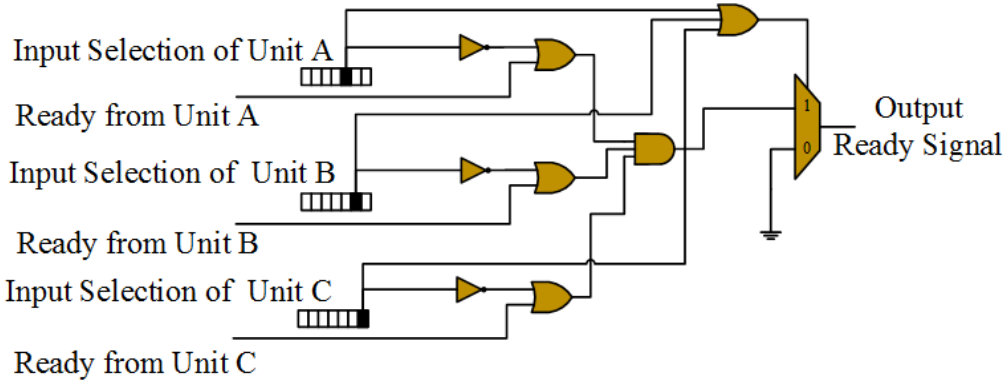
Extra Slides!

Dataflow Control

Buffering System with IFs



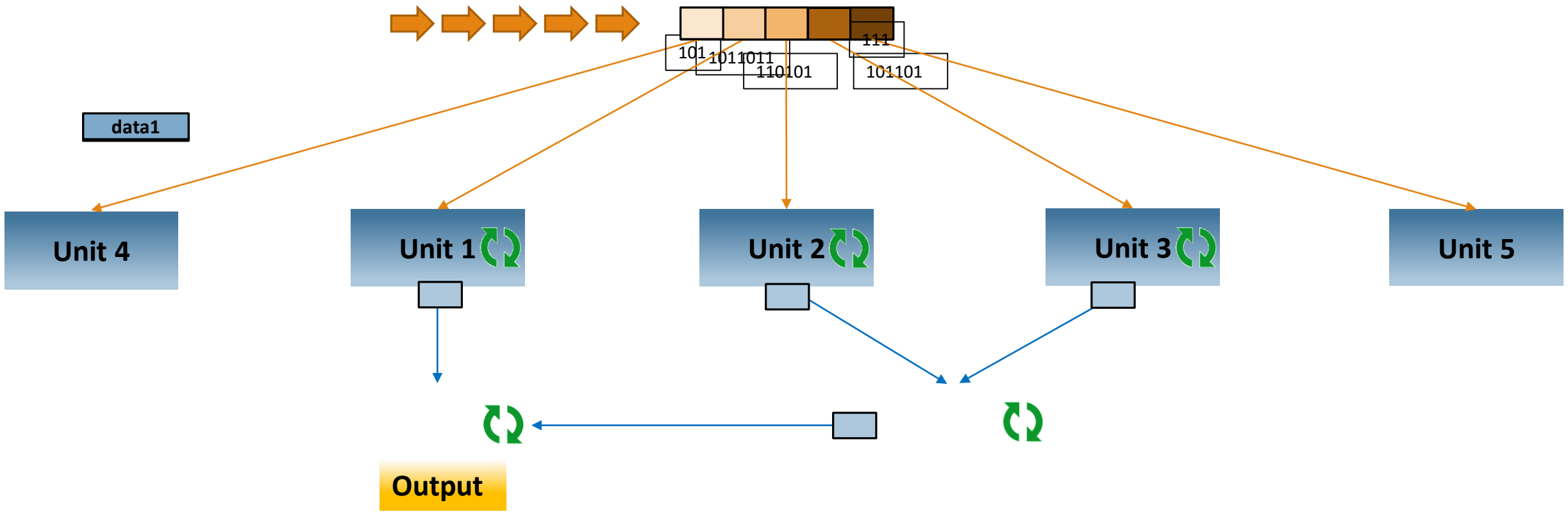
Ready Signal Generation



Suppose there are 3 units A,B,C.
aCb - unit B is configured to receive data from unit A
aR - ready signal from unit A
Ra - ready signal to unit A

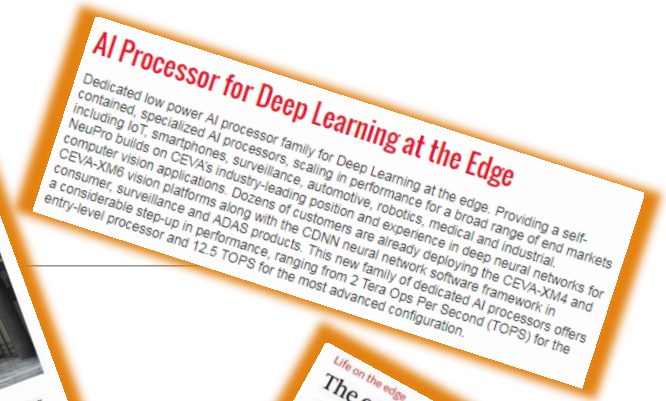
$$Ra = (\sim aCa \vee aR) \wedge (\sim aCb \vee bR) \wedge (\sim aCc \vee cR)$$

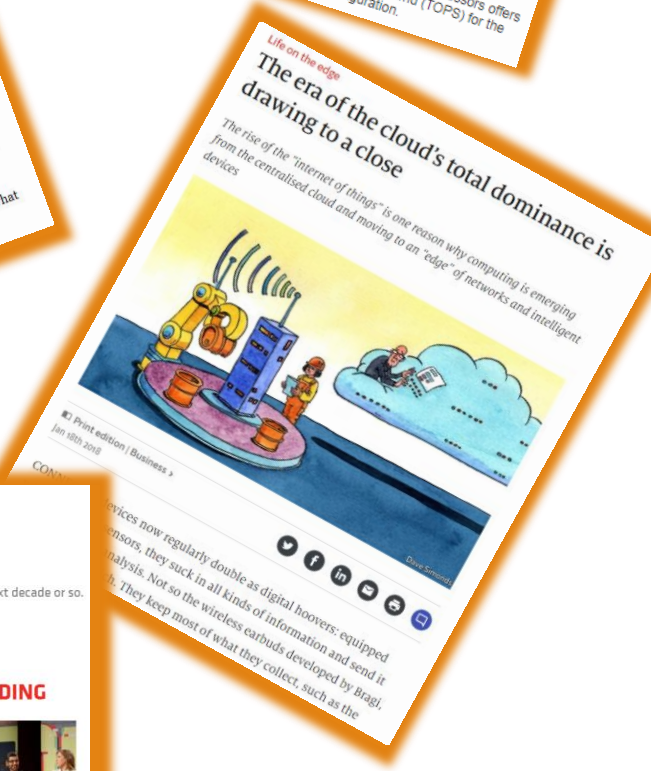
Internal Architecture

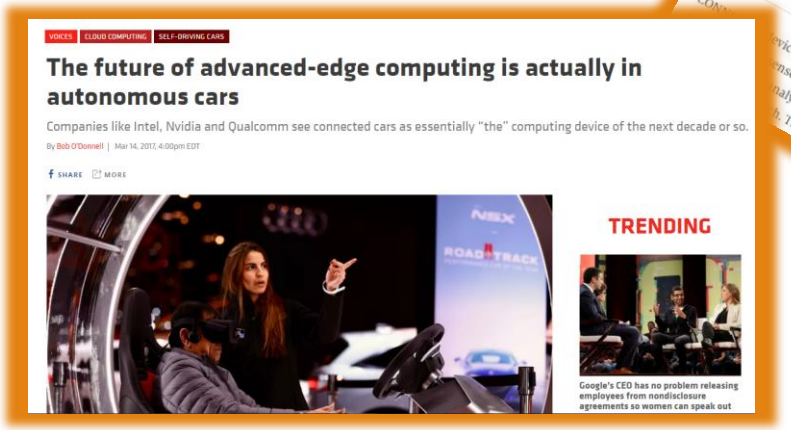


Introduction

- Edge Processing
- Emerging Field
- More Application Areas
- Trends in Vision Industry







Introduction

Edge Processing

- Surveillance
- Autonomous driving
- Robotics