

Performance Design of Streaming Systems

by *Gerrit Muller* HSN-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

Abstract

Video and audio content is a continuous stream of data. Video and audio systems have to be designed in such a way that these streams are processed and delivered continuously. We discuss the pipelining of multiple functions and the impact on bus bandwidth, memory use and CPU overhead.

June 5, 2018

status: preliminary

draft

version: 0

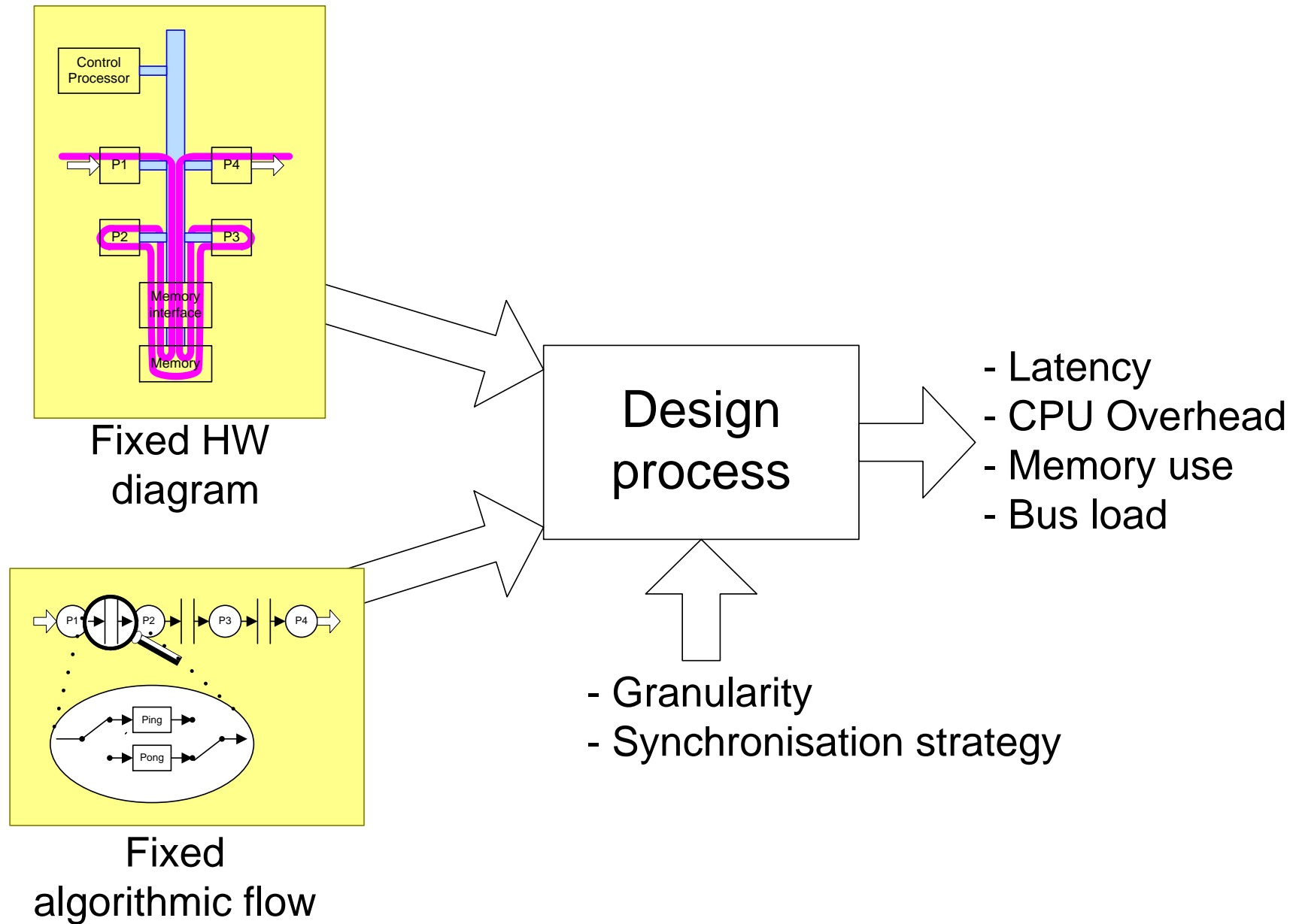
Video Streaming

Hard real-time performance for distributed system with memory-bus

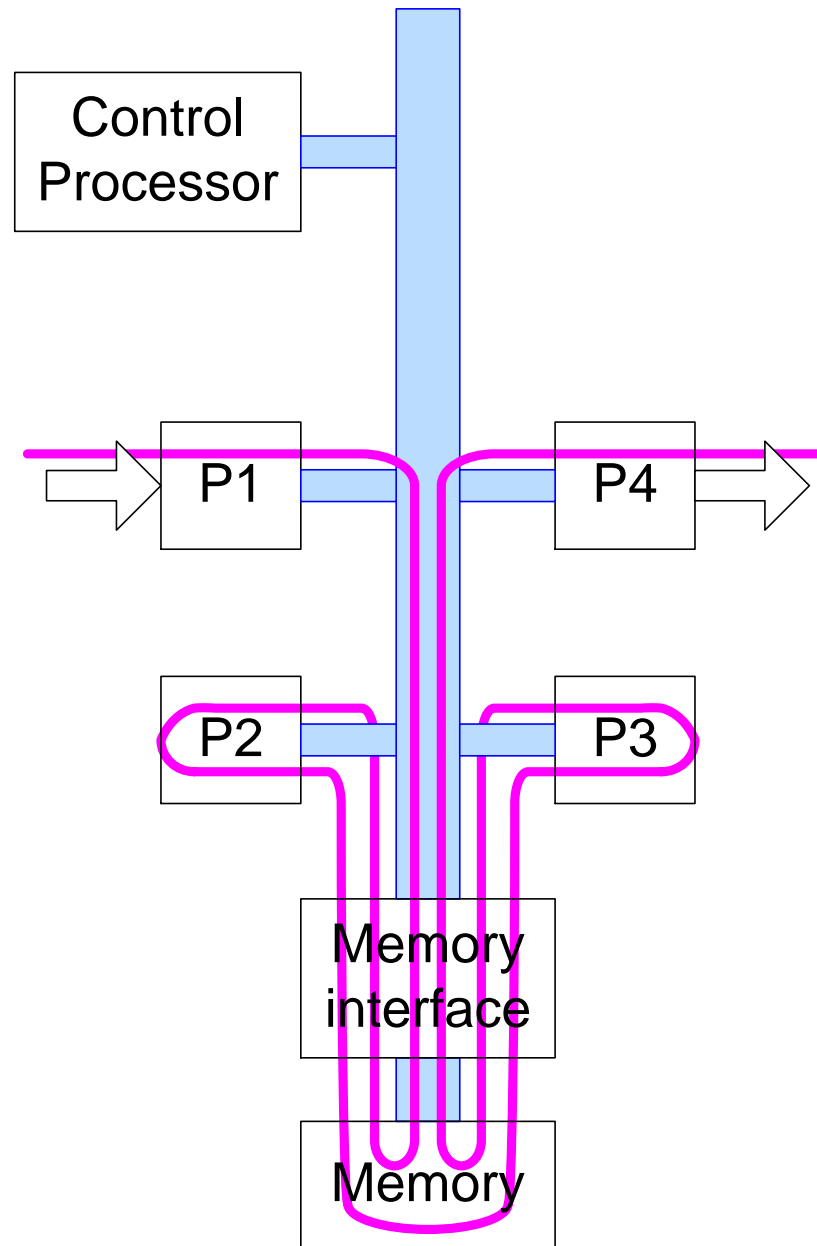
Trade-off of between latency, memory and overhead

Performance consideration in increasing detail

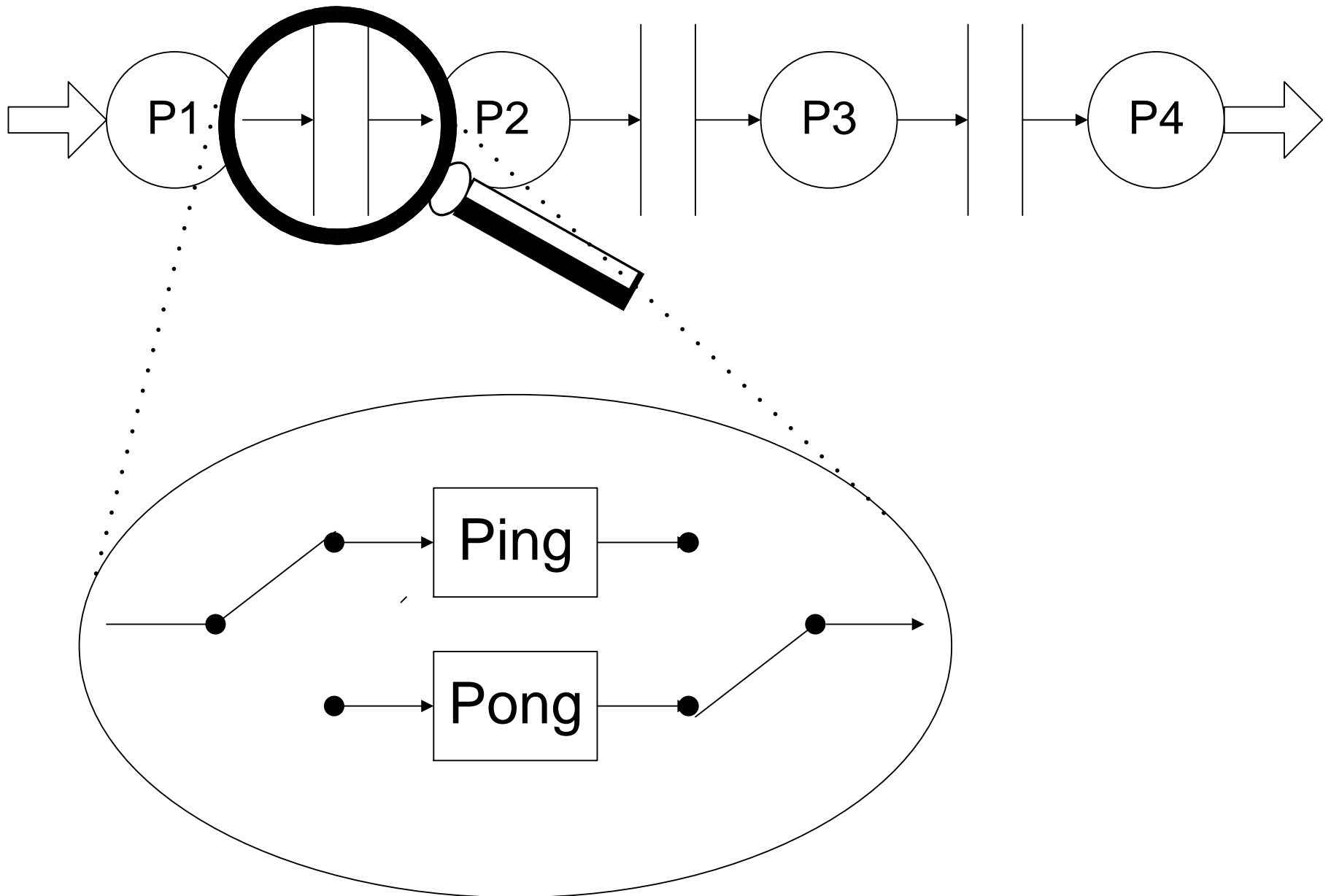
Case Video Streaming: Performance Design



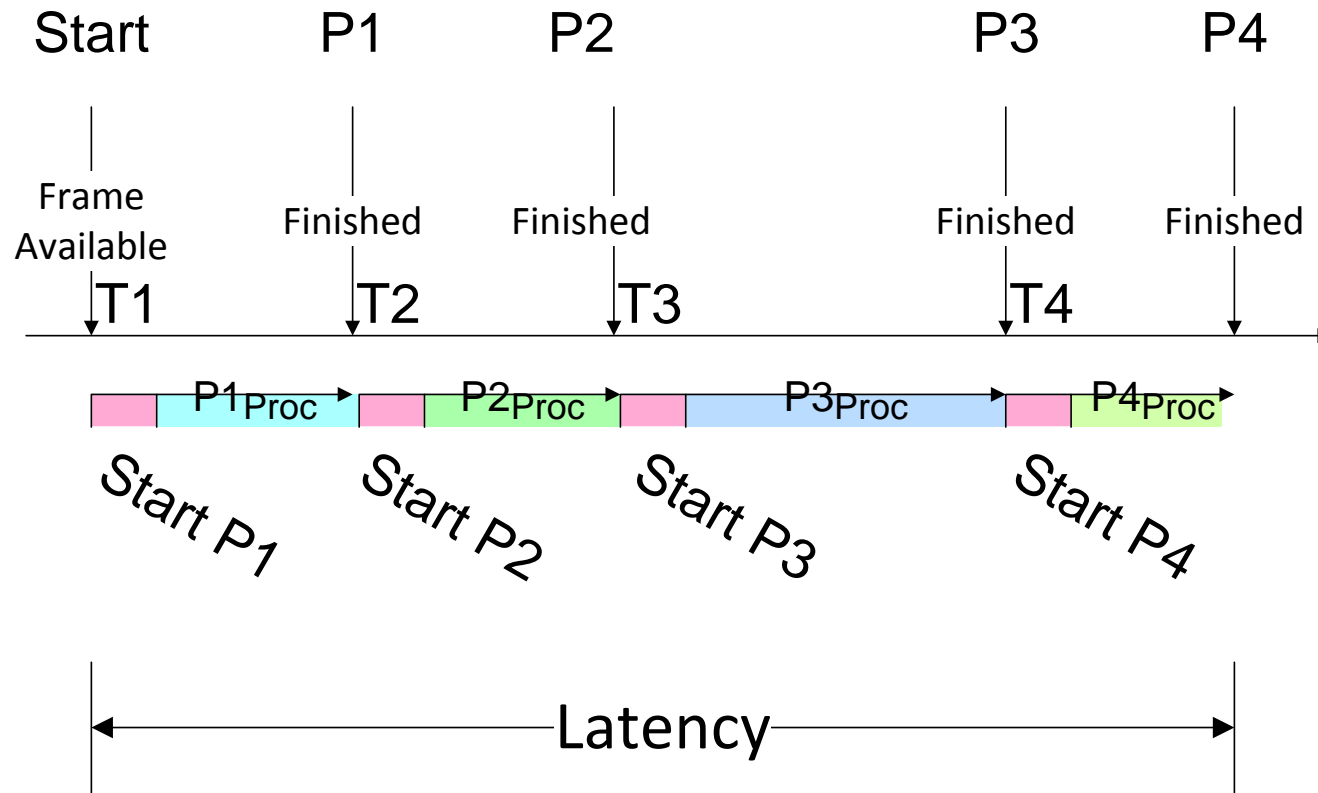
Video Streaming: HW Diagram



Video Streaming Pipeline



Video Streaming: Latency

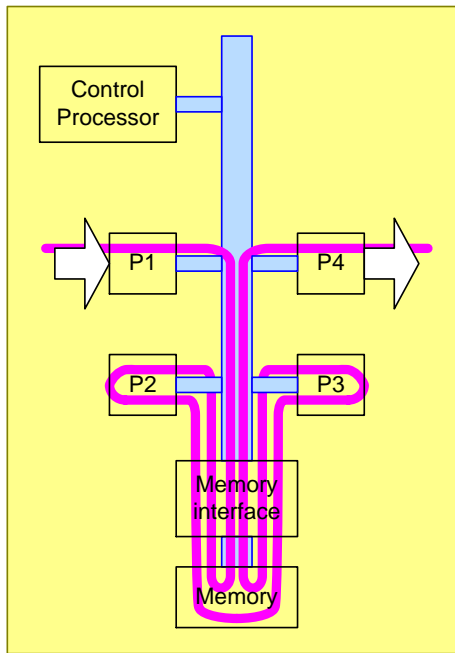


$P3_{Proc} \lesssim \frac{1}{2} T_{frame}$
 $Latency \approx 2 T_{frame}$

Legenda

- Task switch
- Process 1
- Process 2
- Process 3
- Process 4

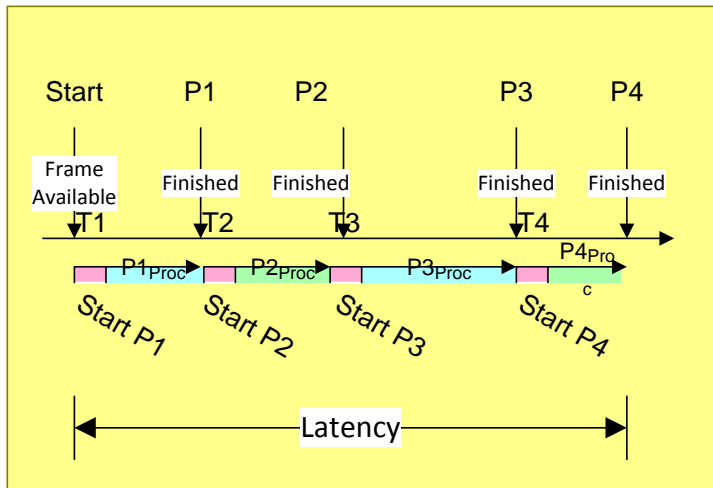
Video Streaming: Resources



$$\text{Overhead} = (T1 + T2 + T3 + T4) * \text{Frame rate}$$

$$\text{Memory usage} = 3 * 2 * \text{Frame size}$$

$$\text{Bus load} = \frac{3 * 2 * \text{Frame size} * \text{Frame rate}}{\text{Bus capacity}} \quad \%$$



T1 .. T4 = Overhead
to start P1 .. P4

Latency Calculation

$$\text{Latency} = \text{Nr. of Proc. blocks}^4 * \text{processing time per block}^{0.01\text{s}} * \text{frame fragment}^1$$

$$\text{Memory} = (\text{Nr. of Proc. blocks}^4 - 1) * 2 * \text{pixels per frame}^{414720} * \text{frame fragment}^1$$

$$\text{Overhead} = \text{Nr. of Proc. blocks}^4 * \text{task switch time}^{10\ \mu\text{s}}$$

$$\text{Overhead (\%)} = \text{Overhead}^{40\ \mu\text{s}} / \text{Latency}^{40\ \text{ms}}$$

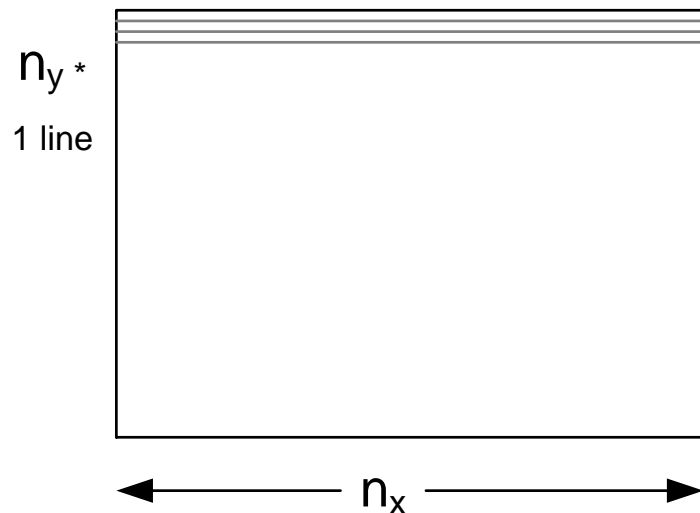
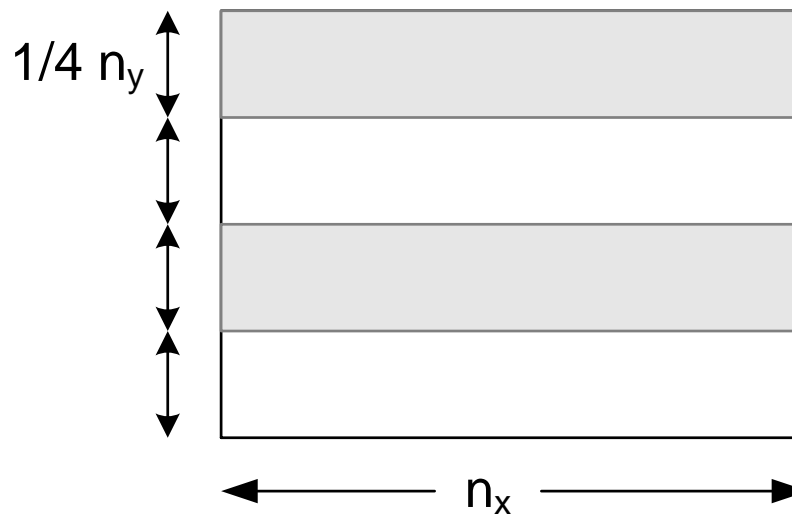
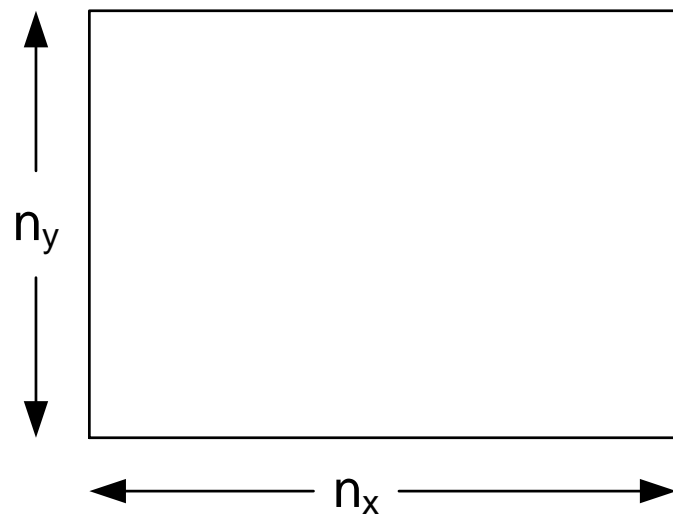
$$\text{Busload} = \text{Memory usage}^{2430\ \text{kB}} * \text{frame fragment}^1 * (\text{frames/s})^{25} / \text{BusCapacity}^{500\text{MB/s}}$$

(mind the units, ms vs. μs and kB vs MB!)

lines	576	pixels per frame	414720
pixels per line	720	Memory in kB	405
		Memory in MB	0.40
frame time	0.04	frame time in μs	40000
task switch time (μs)	10		
Processing per block	0.01	Processing in μs	10000
Bus capacity (MB/s)	500		
Line time (μs)	69		
Frame fragment	Full frame : 1		

Nr of Processing Blocks	4
Latency (ms)	40
Memory (kB)	2430
Overhead (μs)	40
Overhead (%)	0
Busload (%)	12.15

Exercise



Calculate:

Processing time

Overhead

Memory Use

Latency

for buffer size = $1/4$ frame size
and for

buffer size = 1 video line

Exercise Worksheet

Nr of Processing Blocks		4	20
Block size			
	Latency (ms)	40	200
Frame	Memory (kB)	2430	15390
1	Overhead (μ s)	40	200
	Overhead (%)	0	0
	Busload (%)	12.15	76.95
Latency (ms)			
½ Frame	Memory (kB)		
2	Overhead (μ s)		
	Overhead (%)		
	Busload (%)		
Latency (μ s)			
Line	Memory (kB)		
576	Overhead (μ s)		
	Overhead (%)		
	Busload (%)		

lines	576
pixels per line	720
pixels per frame	414720
Memory in kB	405
Memory in MB	0.395508
frame time	0.04
frame time in μ s	40000
task switch time (μ s)	10
Processing per block	0.01
Processing in μ s	10000
Bus capacity (MB/s)	500
Line time (μ s)	69

Changing the Buffer Size

buffer size = 1/4 frame

Processing time = $\frac{1}{4} * \text{original}$ (per fragment)

Latency ~ $\frac{1}{4} * \text{original}$

Overhead = $4 * \text{original}$

Memory use = $\frac{1}{4} \text{ original}$

buffer size = 1 line

Processing time = $\frac{1}{576} * \text{original}$ (per fragment)

Latency ~ $\frac{1}{576} * \text{original} + \text{overhead}$

Overhead = $576 * \text{original}$

Memory use = $\frac{1}{576} \text{ original}$

Video Streaming

Properly designing distributed HRT systems requires trade-off between latency, overhead, and memory needs

Performance model detailing dependent on significance of impact factors

The ASP™ course is partially derived from the EXARCH course developed at *Philips CTT* by *Ton Kostelijk* and *Gerrit Muller*.

Extensions and additional slides have been developed at *ESI* by *Teun Hendriks*, *Roland Mathijssen* and *Gerrit Muller*.