

# From Synchronous to Asynchronous Design

by *Gerrit Muller* Embedded Systems Institute

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

## Abstract

The most simple real time programming paradigm is a synchronous loop. This is an effective approach for simple systems, but at a certain level of concurrent activities an asynchronous design, based on scheduling tasks, becomes more effective. We will use a conventional television as case to show real time design strategies, starting with a straightforward analog television based on a synchronous design and incrementally extending the television to become a full-fledged digital TV with many concurrent functions.

The complete course ASP<sup>TM</sup> is owned by Embedded Systems Institute. To teach this course a license from Embedded Systems Institute is required. This material is preliminary course material. The final material and course information can be found at: [www.esi.nl/cursus](http://www.esi.nl/cursus).

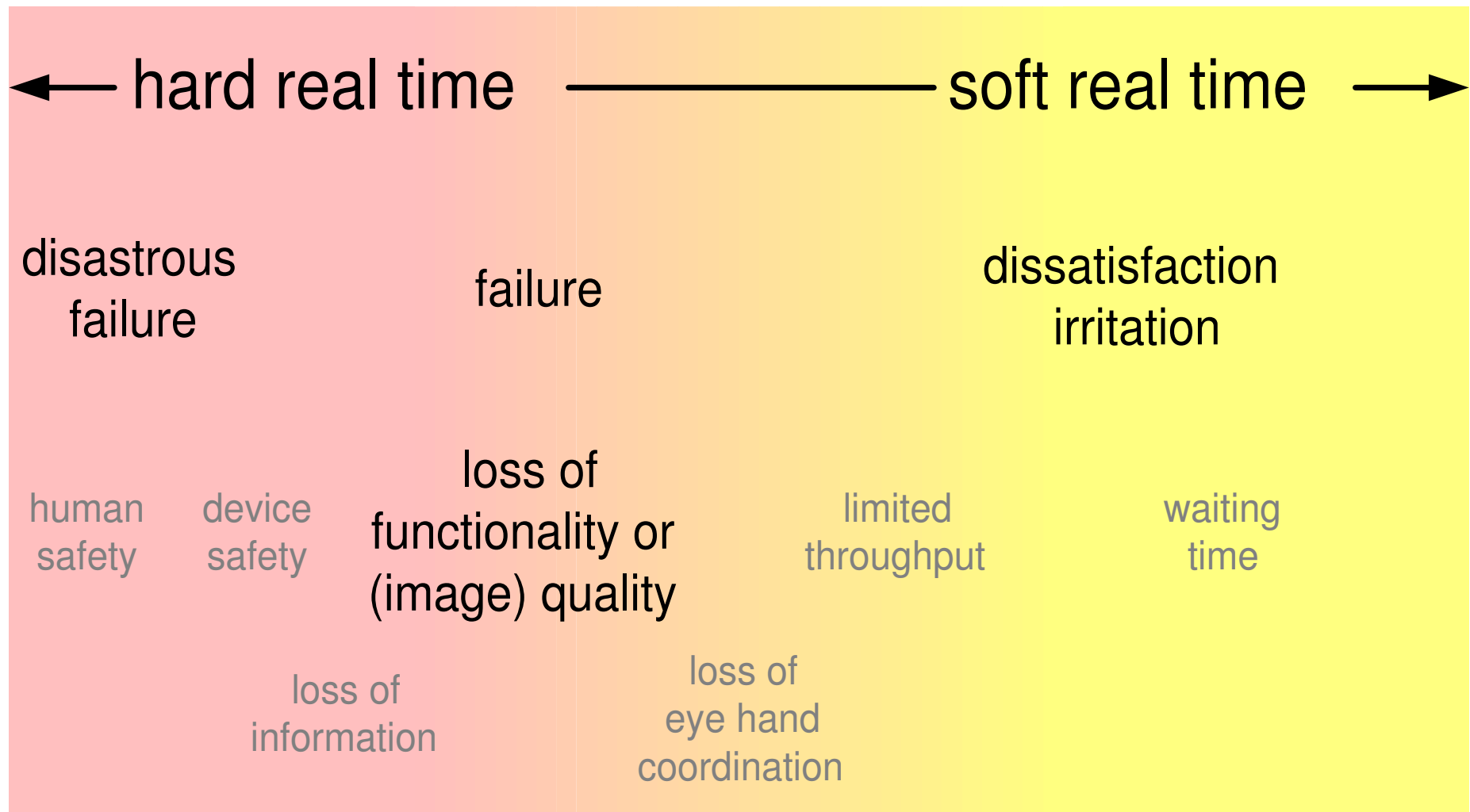
February 10, 2011

status: preliminary

draft

version: 0

# Hard Real Time Design



## *Simple Analog TV*

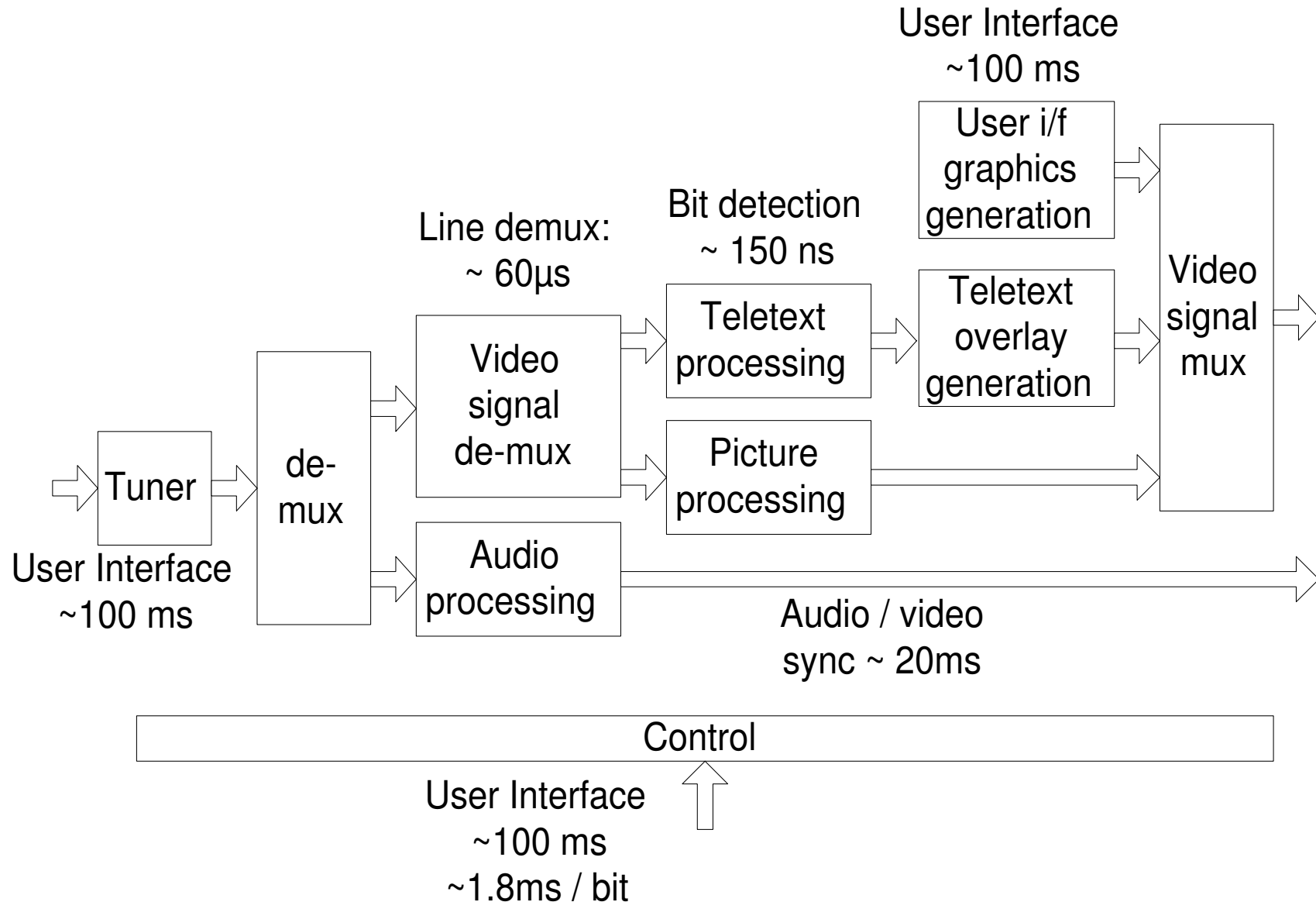
Multiple views on system

Fundamentals of *periodic* or *streaming* Hard Real-Time applications

System performance characterisation: Performance model

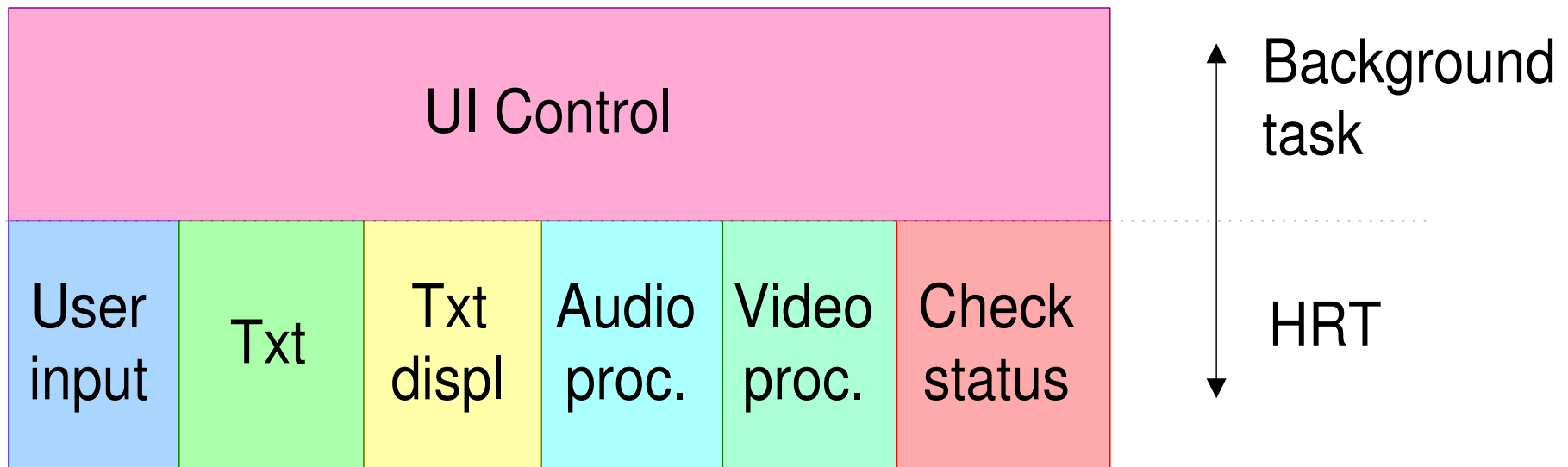
Synchronous design concept

# Functional Flow Simple Analog Television

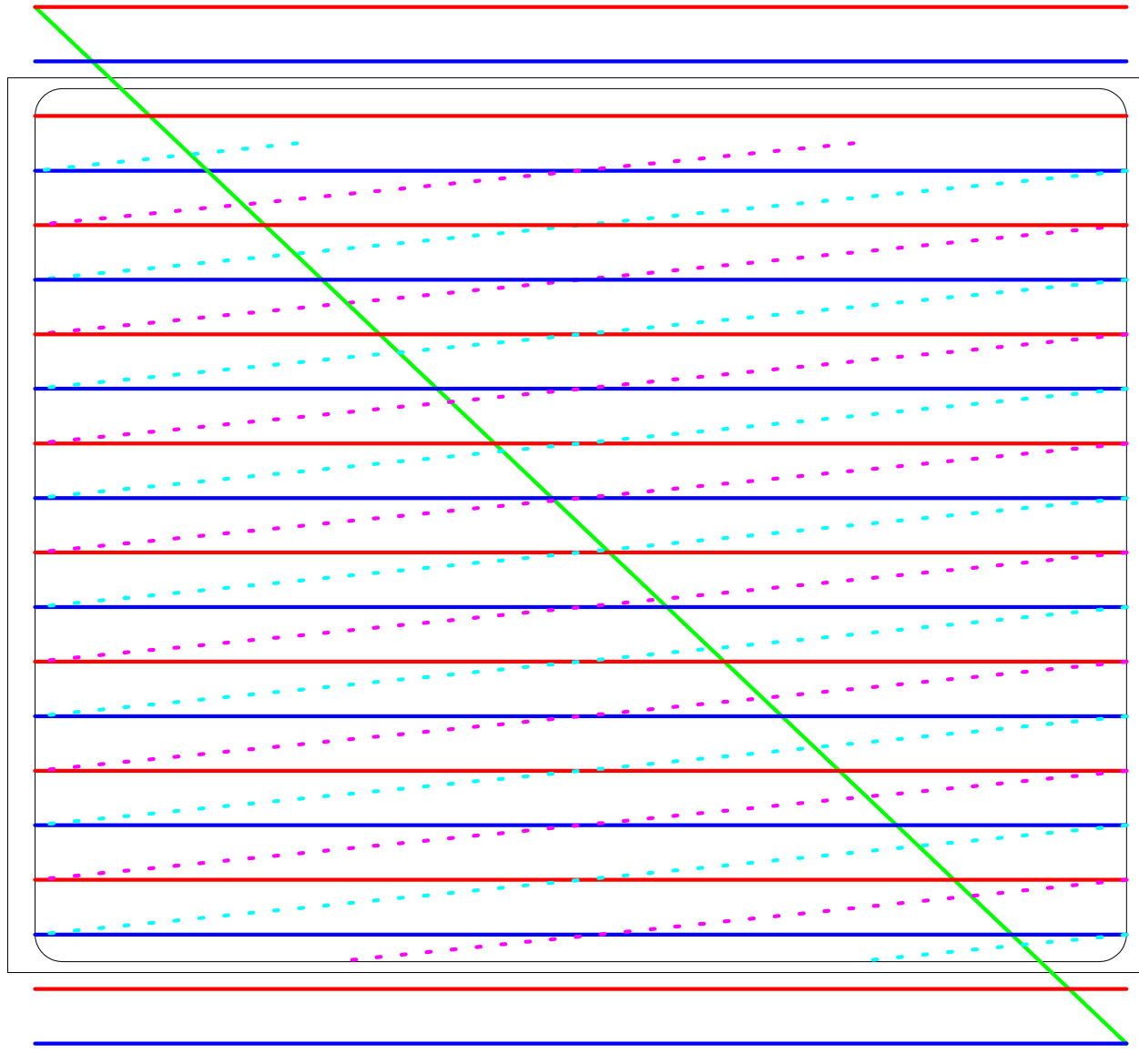


# SW Construction Diagram

---



# Video Timing



Hidden lines  
(can contain data)

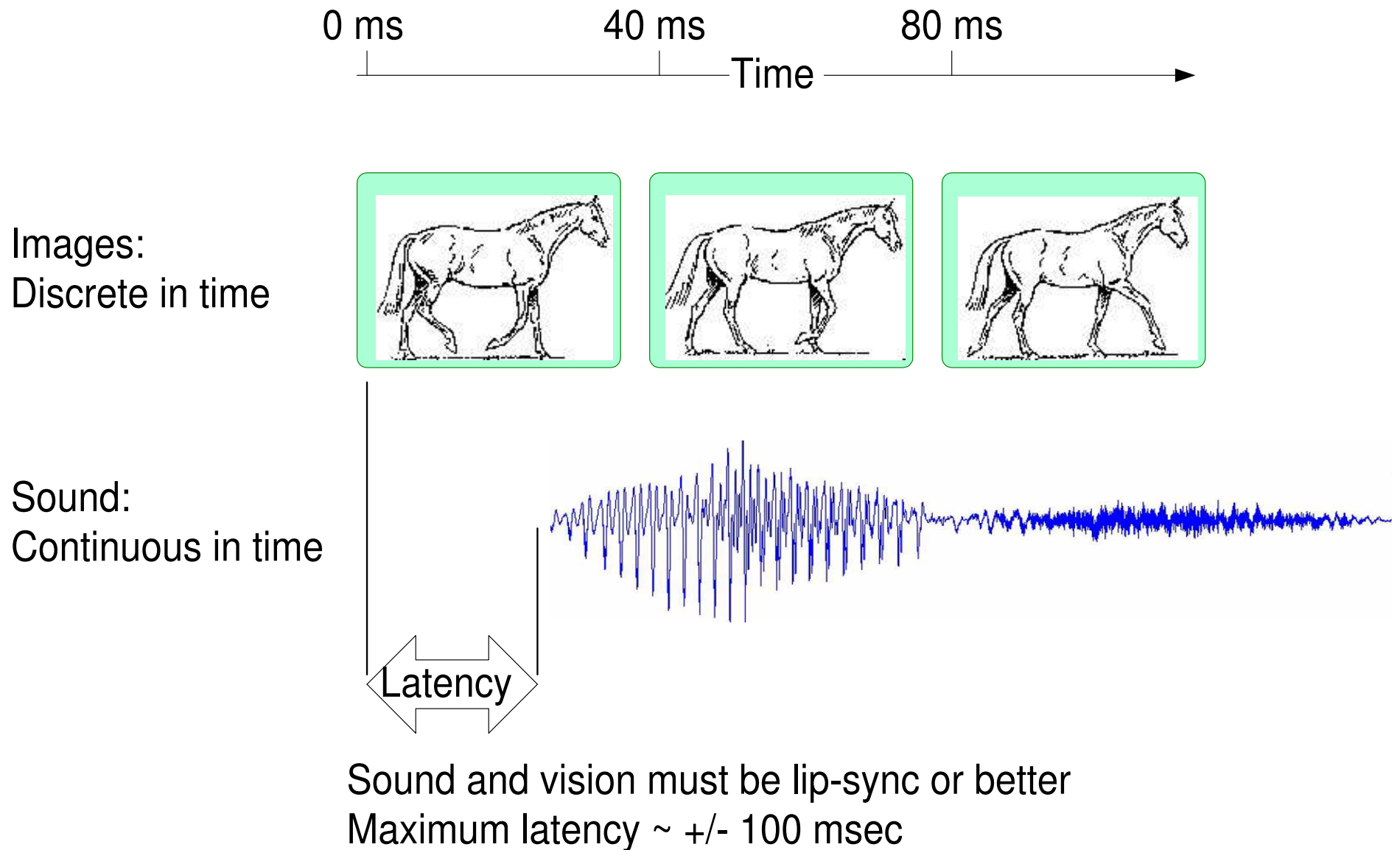
- Scan line even
- Scan line odd
- - - Retrace even
- - - Retrace odd
- Vertical retrace

For PAL-625:

Line Frequency: 15.625 kHz  
Scanning Lines: 625  
Field Frequency: 50 Hz

Hidden lines  
(can contain data)

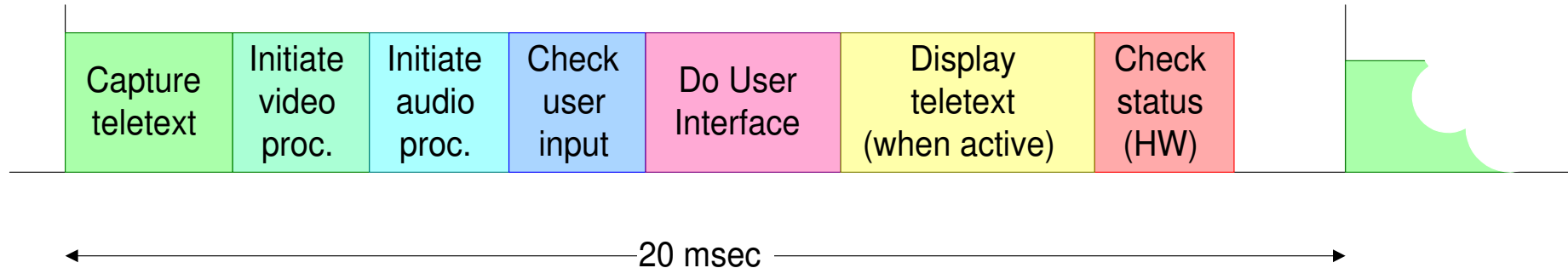
# Audio-Video Synchronization Requirement



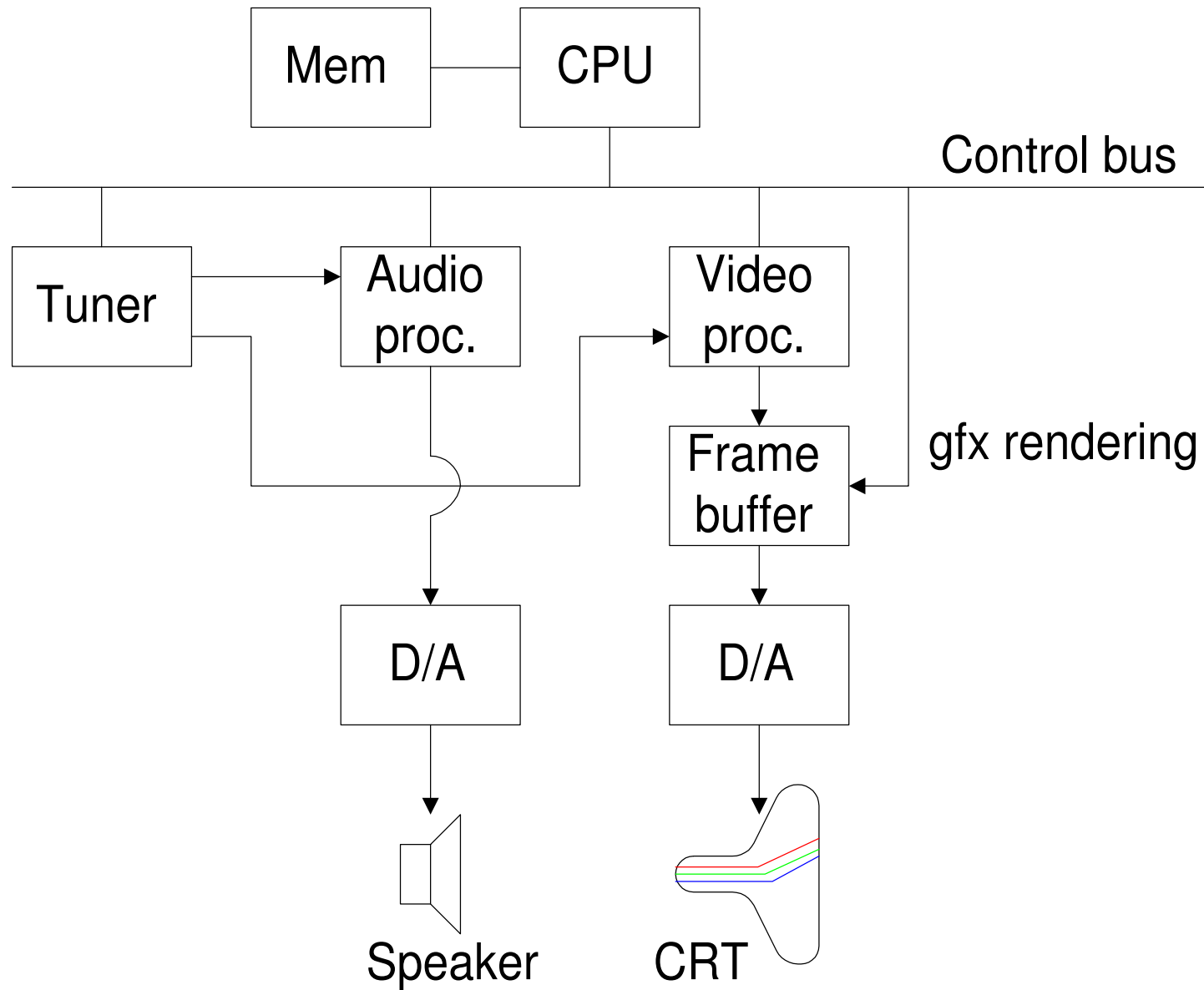
## Synchronous design

Frame  
interrupt

Frame  
interrupt



# HW Diagram



## Synchronous design questions

Estimate processing time on a 100 MHz ARM core

Assuming that all processing and acquisition is done in HW

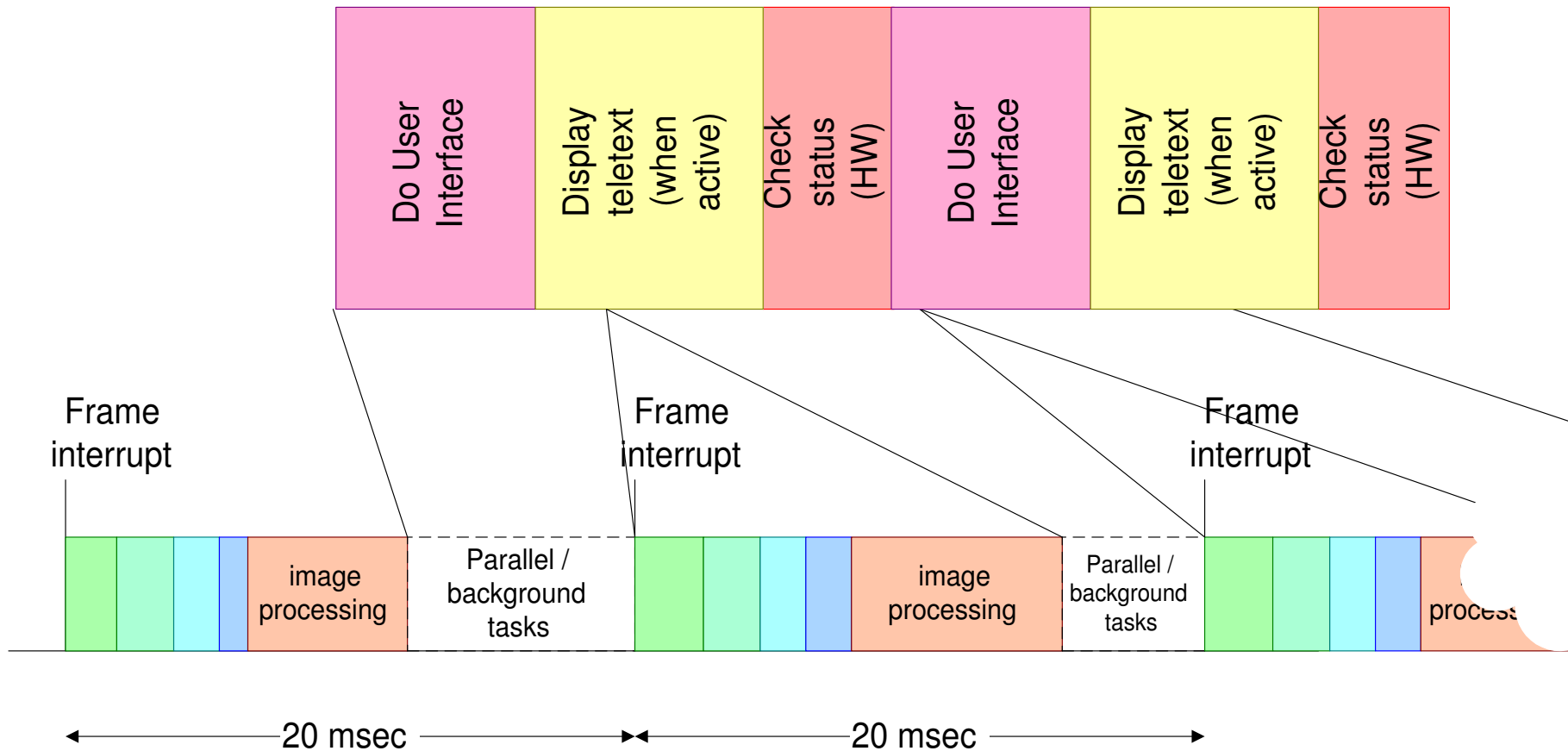
Graphics rendering (user interface + teletext display) is done in SW

Where do you expect variation?

How feasible and how reliable is this design?

# Low Priority Work in the Background

## Design with multiple parallel tasks



# Synchronous or Asynchronous?

---

## Synchronous

=> Map on Highest frequency

### Constraints:

- Processing frequency must be a whole (integer) multiple of the lower frequencies
- Each process must be completed within the period of the highest frequency, together with the high-frequency process

## A-Synchronous

=> Concurrent processes

# Multiple Periods in a Simple TV

---

Input signal	50 Hz
Processing	100 Hz
User Interface	20 Hz
Power and Housekeeping	0.5 Hz
Output	50, 100 Hz

# Summary Case Simple Analog TV

---

## *Simple Analog TV*

Performance model requires:

- identification of processing steps
- their relation
- critical parameters and values

Synchronous design sufficient for periodic applications with one dominant frequency

Multiple views on system:

- HW diagram
- SW construction diagram
- Functional flow
- Time-line

## *From Analog TV to Digital TV*

Adding more input formats and output devices

Multiple heterogenous periods: asynchronous design with concurrent tasks.

Input                      Many frequencies  
Video & Audio variable timing

Output                    Many frequencies

Processing              Variable

Many video variants (see table)

Many audio variants (quality, number of speakers, ...)

# Simple Video Processing Pipeline

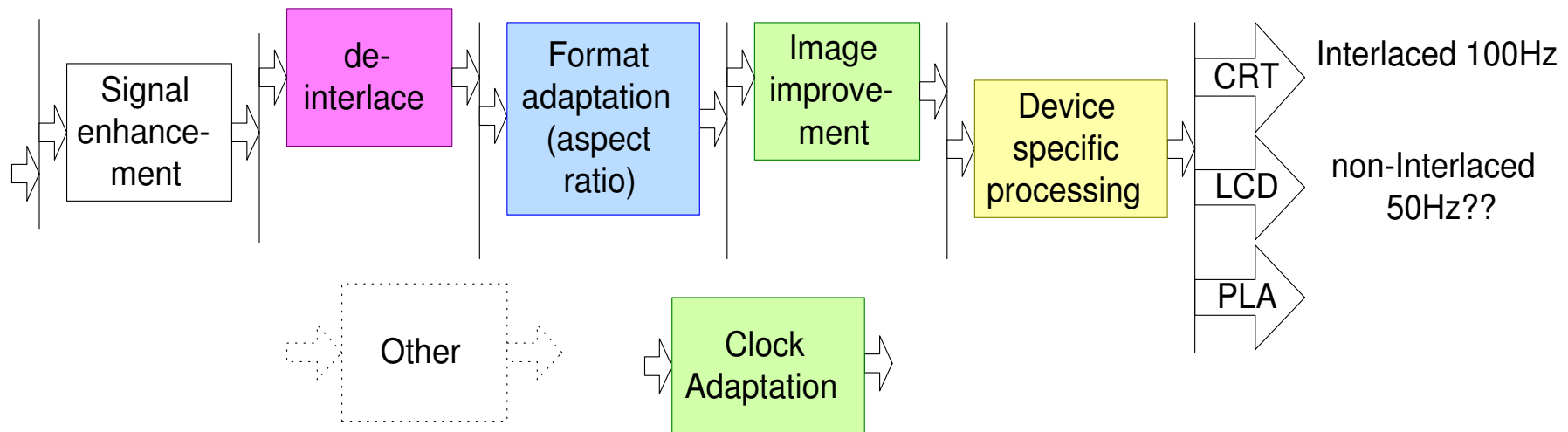
## multi task design complex TV

In modern television the format of the image can change (e.g. widescreen)

The user can set the refresh rate to higher values (e.g. 100Hz anti-flicker)

Different displays (CRT, LCD, Plasma) can be attached that need the image in different formats (interlaced, non-interlaced, different refresh rates)

Non interlaced images need special filtering of the image to prevent ragged images



# Table with ATSC Video Formats

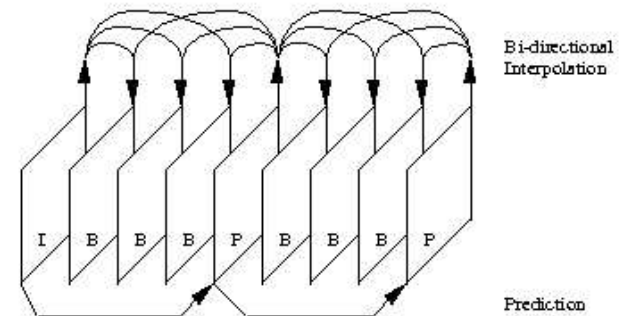
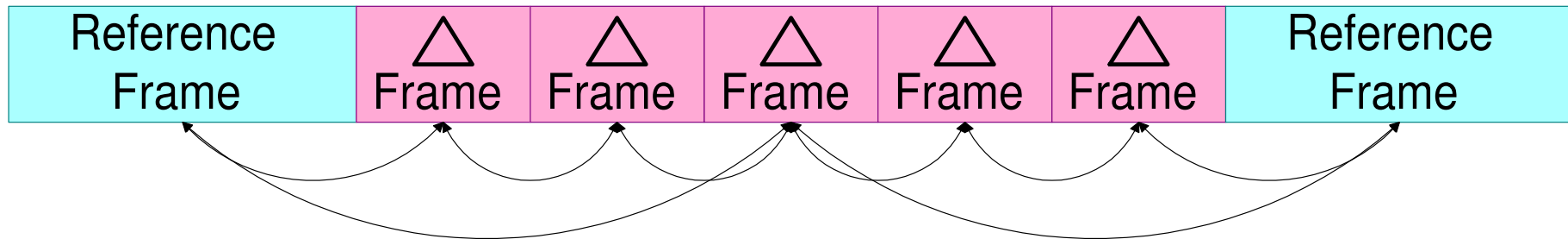
spec	Horizontal pixels	Vertical pixels	Aspect ratio	Monitor interface	Format name	Frames per sec	Fields per sec	Transmitted interlaced
					1080i60	30	60	yes
	1920	1080	16:09	1080i	1080p30	30	30	no
					1080p24	24	24	no
					720p60	60	60	no
	1280	720	16:09	720p	720p30	30	30	no
					720p24	24	24	no
				480p	480p60	60	60	no
	704	480	16:09		480i60	30	60	yes
				480i	480p30	30	30	no
ATSC					480p24	24	24	no
				480p	480p60	60	60	no
	704	480	04:03		480i60	30	60	yes
				480i	480p30	30	30	no
					480p24	24	24	no
				480p	480p60	60	60	no
	640	480	04:03		480i60	30	60	yes
	640			480i	480p30	30	30	no
					480p24	24	24	no
NTSC	»640	483	04:03	Note 1	Note 1	30	60	yes
Note 1: Some people refer to NTSC as 480i.								

Source: [http://www.hdtvprimer.com/ISSUES/what\\_is\\_ATSC.html](http://www.hdtvprimer.com/ISSUES/what_is_ATSC.html)

# Data Packets in Digital TV



Packet



## *From Analog TV to Digital TV*

Real-life applications rapidly introduce all kinds of variations  
Concurrent tasks cope with different periods

The ASP <sup>TM</sup> 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* .