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.
Hard Real Time Design

- **Hard Real Time**
  - Disastrous failure
  - Human safety
  - Device safety
  - Loss of information

- **Soft Real Time**
  - Dissatisfaction
  - Limited throughput
  - Waiting time
  - Loss of functionality or (image) quality
  - Loss of eye hand coordination

From Synchronous to Asynchronous Design

©2006, Embedded Systems Institute

©2006, Gerrit Muller

version: 0
March 6, 2013
PHRTpositioning
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

User Interface
~100 ms

Video signal de-mux

Audio processing

Bit detection
~ 150 ns

Teletext processing

Teletext overlay generation

User i/f graphics generation

Video signal mux

Audio / video sync ~ 20ms

Control

User Interface
~100 ms
~1.8ms / bit

From Synchronous to Asynchronous Design
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0
March 6, 2013
PHRTTelevisionArchitectureSimple
Video Timing

For PAL-625:
- Line Frequency: 15.625 kHz
- Scanning Lines: 625
- Field Frequency: 50 Hz

Hidden lines (can contain data)

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

Hidden lines
(can contain data)
Audio-Video Synchronization Requirement

Images:
Discrete in time

Sound:
Continuous in time

Latency
Sound and vision must be lip-sync or better
Maximum latency ~ +/- 100 msec
Synchronous Control Software

Synchronous design

Frame interrupt

Capture teletext
Initiate video proc.
Initiate audio proc.
Check user input
Do User Interface
Display teletext (when active)
Check status (HW)

Frame interrupt

20 msec
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

- Do User Interface
- Display teletext (when active)
- Check status (HW)
- Do User Interface
- Display teletext (when active)
- Check status (HW)

Frame interrupt

- image processing
- Parallel / background tasks
- image processing
- Parallel / background tasks

20 msec

From Synchronous to Asynchronous Design
©2006, Embedded Systems Institute
©2006, Gerrit Muller
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

<table>
<thead>
<tr>
<th>Category</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input signal</td>
<td>50 Hz</td>
</tr>
<tr>
<td>Processing</td>
<td>100 Hz</td>
</tr>
<tr>
<td>User Interface</td>
<td>20 Hz</td>
</tr>
<tr>
<td>Power and Housekeeping</td>
<td>0.5 Hz</td>
</tr>
<tr>
<td>Output</td>
<td>50, 100 Hz</td>
</tr>
</tbody>
</table>
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.
### Digital Television

<table>
<thead>
<tr>
<th>Input</th>
<th>Many frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Video &amp; Audio variable timing</td>
</tr>
<tr>
<td>Output</td>
<td>Many frequencies</td>
</tr>
<tr>
<td>Processing</td>
<td>Variable</td>
</tr>
</tbody>
</table>

Many video variants (see table)
Many audio variants (quality, number of speakers, ...)

From Synchronous to Asynchronous Design
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0
March 6, 2013
PSRTdigitalTV
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>
<thead>
<tr>
<th>spec</th>
<th>Horizontal pixels</th>
<th>Vertical pixels</th>
<th>Aspect ratio</th>
<th>Monitor interface</th>
<th>Format name</th>
<th>Frames per sec</th>
<th>Fields per sec</th>
<th>Transmitted interlaced</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATSC</td>
<td>1920</td>
<td>1080</td>
<td>16:09</td>
<td>1080i</td>
<td>1080i60</td>
<td>30</td>
<td>60</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1080p30</td>
<td>30</td>
<td>30</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1080p24</td>
<td>24</td>
<td>24</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>1280</td>
<td>720</td>
<td>16:09</td>
<td>720p</td>
<td>720p60</td>
<td>60</td>
<td>60</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>720p30</td>
<td>30</td>
<td>30</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>720p24</td>
<td>24</td>
<td>24</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>704</td>
<td>480</td>
<td>16:09</td>
<td></td>
<td>480p60</td>
<td>60</td>
<td>60</td>
<td>no</td>
</tr>
<tr>
<td>ATSC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480i60</td>
<td>30</td>
<td>60</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480p30</td>
<td>30</td>
<td>30</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480p24</td>
<td>24</td>
<td>24</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>704</td>
<td>480</td>
<td>04:03</td>
<td></td>
<td>480p60</td>
<td>60</td>
<td>60</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480i60</td>
<td>30</td>
<td>60</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480p30</td>
<td>30</td>
<td>30</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480p24</td>
<td>24</td>
<td>24</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>480</td>
<td>04:03</td>
<td></td>
<td>480p60</td>
<td>60</td>
<td>60</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480i60</td>
<td>30</td>
<td>60</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>480</td>
<td>04:03</td>
<td></td>
<td>480p30</td>
<td>30</td>
<td>30</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>480p24</td>
<td>24</td>
<td>24</td>
<td>no</td>
</tr>
<tr>
<td>NTSC</td>
<td>≈640</td>
<td>483</td>
<td>04:03 Note 1</td>
<td>Note 1</td>
<td>Note 1</td>
<td>30</td>
<td>60</td>
<td>yes</td>
</tr>
</tbody>
</table>

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

Reference Frame

Frame

Frame

Frame

Frame

Frame

Reference Frame

Bi-directional
Decompression
From Analog TV to Digital TV

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