PONG

Charles Hastings
Rachid Jeitani

Embedded Systems Design Project
Overview

• Classic Pong without so much socializing
• Each player controls a paddle using a mouse
• One paddle per monitor
• Uses network to coordinate scores, serving and balls going back and forth
Block Diagram
Mouse

• PS/2 mouse protocol is rather unstructured
  – No complex packet structure, just a few bytes
  – Reliable legacy hardware is sparse

• Interrupt on data in FIFO, ISR integrates

• Polling interface also implemented in attempt to improve system stability
Ethernet

• 100Mbit=fast, 16-bit ISA=slow
  – Interface appears to limit 30Hz status updates
  – IP and UDP removed to reduce overheard
  – DMA is the next step

• Polling interface implemented, too

• Needs a clock! And power-on reset
VGA

• Three main visual layers
  – Background (16x16 bitmap)
  – Game Objects (Paddles & Ball)
  – Foreground (320x240 bitmap)

• Uses two different memories to store data
Storing Graphics

• RAM for Game Objects and Background
  – 26 elements, each 16-bits wide
  – Stores the 16x16 background tile
  – Stores paddle & ball positions
  – Stores colors

• Foreground RAM
  – 4800 elements, each 16-bits wide
  – Stores ¼ of the screen which is then stretched to fill it
VGA Block Diagram
Lessons Learned

• Integration of working parts can still be time-consuming

• Tools are your best friend and your worst enemy

• Plan hardware to the last detail. Trial and error is very time consuming

• Immediate success takes the fun out of it
Tools

- Quartus, SOPC Builder, NiOS
- Wireshark (packet capture)
- BitTwister (win32 pcap replay)
- Subversion
- GIMP (for XBM arrays)
- NAFE (font editor)