Electronic Etch-a-Sketch Project

- Project Concept and Background
- Checkpoint Structure
- Bells and Whistles

Objectives

- Broad "brush" overview of the project
- Details will be covered in the lab lectures
- NOTE: anything discussed in the lab lectures and project checkpoint write-ups supercedes what I describe here!
  - The TAs know the project better than I do!
**Electronic Etch-a-Sketch**

- AKA an electronic "paint" machine ...
- Recent CS 150 projects: network streaming audio, network digital "telephone", pong, video image processing, ... audio and/or video a common theme
- This is a good CS 150 project because ...
  - It is NOT a processor architecture (take CS 152 for that!)
  - Vast majority of digital designs involve interfacing with the real world, with real world timing constraints!
  - Hardware Systems: Input, Processing, Output
    - Pen/brush position/size (and special effects) inputs
    - State of the screen in frame buffer in SDRAM
    - Modified by drawing semantics
    - Refresh LCD screen from frame buffer in SDRAM

---

**Etch-a-Sketch**
Etch-a-Sketch

One possible way to do it ...

Electronic Etch-a-Sketch

Implement this ... with this ...
Calinx Board

- Video & Audio Ports
- AC '97 Codec & Power Amp
- Video Encoder & Decoder
- Flash Card & Micro-drive Port
- Prototype Area
- Seven Segment LED Displays
- Four 100 Mb Ethernet Ports
- 8 Meg x 32 SDRAM
- Quad Ethernet Transceiver
- Xilinx Virtex 2000E

Etch-a-Sketch Block Architecture

Checkpoint #1: N64 Game Controller Interface
Checkpoint #4: Paint Engine Interface
- Cursor Position
- Cursor Size
- Brush Color

Checkpoint #2: NTSC Video
Checkpoint #3: SDRAM
- Controller
- Arbiter
Checkpoint #1: N64 Interface

- This week in lab!
- Continually poll N64 and report state of buttons and analog joystick
  - Issue 8-bit command
  - Receive 32-bit response
- Each button response is 32 bit value containing button state and 8-bit signed horizontal and vertical velocities
- Serial interface protocol
  - Multiple cycles to perform each transaction
- Bits obtained serially--UART-like functionality
  - Framing (packet start/stop)
  - Bit encoding
    - start | data | data | stop

Checkpoint #2: Video Encoding

- Pixel Array:
  - Digital image represented by matrix of values, where each is a function of the information surrounding it in the image; single element in image matrix: picture element or pixel (includes info for all color components)
  - Array size varies for different apps and costs: some common sizes shown
- Frames:
  - Illusion of motion created by successively flashing still pictures called frames
Checkpoint #2: Video Encoding

- Video details fairly complex and involve many choices:
  - NTSC vs. PAL, HDTV, ...
  - Interleaved even-odd frames (TV) vs. progress scan (computer and digital displays)
  - Frame size, frame rate
  - Pixel encodings: RGB, YUV/YCB (Luminance, Chrominance -- brightness plus color difference signals)
  - Subsampling to reduce data demands (compression trick)
  - Inputs: ITU-R BT.601 Format (Digital Broadcast NTSC)
  - Outputs: Component video, S-video to drive LCDs in lab
  - Fortunately, Calinx board has a chip on-board that deals with much of the grungy details ...

---

**ITU-R BT.656 Details**

- Interfacing details for ITU-601
  - Pixels per line: 858
  - Lines per frame: 525
  - Frames/sec: 29.97
  - Pixels/sec: 13.5 M
  - Viewable pixels/line: 720
  - Viewable lines/frame: 487

- With 4:2:2 chroma sub-sampling, send 2 words/pixel (Cr/Y/Cb/Y)
- Words/sec = 27M
- Encoder runs off a 27MHz clock
- Control info (horizontal & vertical synch) is multiplexed on data lines
- Encoder data stream show to right
- See video tutorial documents on course documentation web page

---

**Note**: Interfacing details in parentheses are for 525-line systems which differ from those for 625-line systems. (See also Recommendation ITU-R BT.601.)
Checkpoint #2: Video Encoder

- Paint engine processes pixels within frame buffer
- Drive ADV7194 video encoder device to output correct NTSC video
- Gain lots of experience reading data sheets
- Dictates the 27 MHz operation rate
  - Used throughout graphics subsystem

---

Calinx On-Board Video Encoder

- Analog Devices ADV7194: ITU 601/656 in, Composite Video Out

- Supports:
  - Multiple input formats and outputs
  - Operational modes, slave/master
  - Used in default mode: ITU-601 as slave s-video output

- Digital input side connected to Virtex pins
- Analog output side wired to on board connectors or headers
- I2C interface for initialization:
  - Wired to Virtex
Checkpoint #3: SDRAM Interface

- Memory protocols
  - Bus arbitration
  - Address phase
  - Data phase
- DRAM is large, but few address lines and slow
  - Row & col address
  - Wait states
- Synchronous DRAM provides fast synchronous access current block
  - Little like a cache in the DRAM
  - Fast burst of data
- Arbitration for shared resource

SDRAM READ Burst Timing
Checkpoint #4: Paint Engine

Fed a series of "brush" strokes and effects
- Defined by color (including erase), brush shape (height, width), etc.
- Current brush positions
- All from N64 Controller interface
- NOTE: lots of buttons, and lots of opportunity for extra credit brush effects

Renders "paint" into frame buffer within range of pixels at current cursor position

Must arbitrate for SDRAM and carry out bus protocol
- Paint engine overlaps writing SDRAM with video interfacing reading SDRAM to drive LCD display
- Continuously feeding video interface from SDRAM subject to timing specifications

Checkpoint Structure and Schedule

- Checkpoint #1: Game Controller Interface (Week 7)
- Checkpoint #2: LCD Interface (Week 8)
- Checkpoint #3: Memory Interface (Week 9, 10 -- Spans Midterm II)
- Checkpoint #4: Paint Engine/Integration (Week 11, 12, 13+)
- Early Check-off: Wednesday before Thanksgivings (November 23)
- Standard Check-off: Wednesday, November 30
- Project Final Report: December 7
Possible Bells and Whistles

- Still thinking about the bells and whistles.
  These are mostly implemented in the paint engine.
  - Brush sound effects
  - Brush effects, like spray varn, sprays, etc.
  - Brush shapes, like square, oval, triangle
  - Brush physics, soft vs hard brush, brush angle, etc. (e.g., Japanese calligraphy)
  - Color mixing rather than overwrite
  - Your good idea here

NOTE: We don’t necessarily know how to implement these ourselves!
NOTE: Extra credit will be limited to 20% extra points and no extra credit unless the standard functionality works.