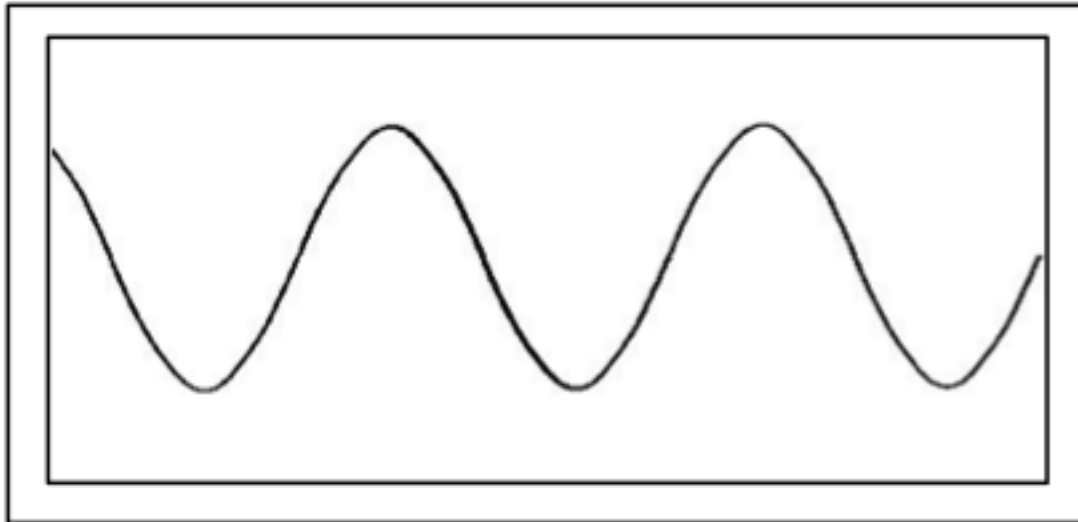


# A Controllable Function Generator

Sarah Ferguson  
Gavin Darcey  
Fall 2010

# Overview



Amplitude: 3 V

Frequency: 100 Hz

Duty Cycle: N/A %

Square



Sine



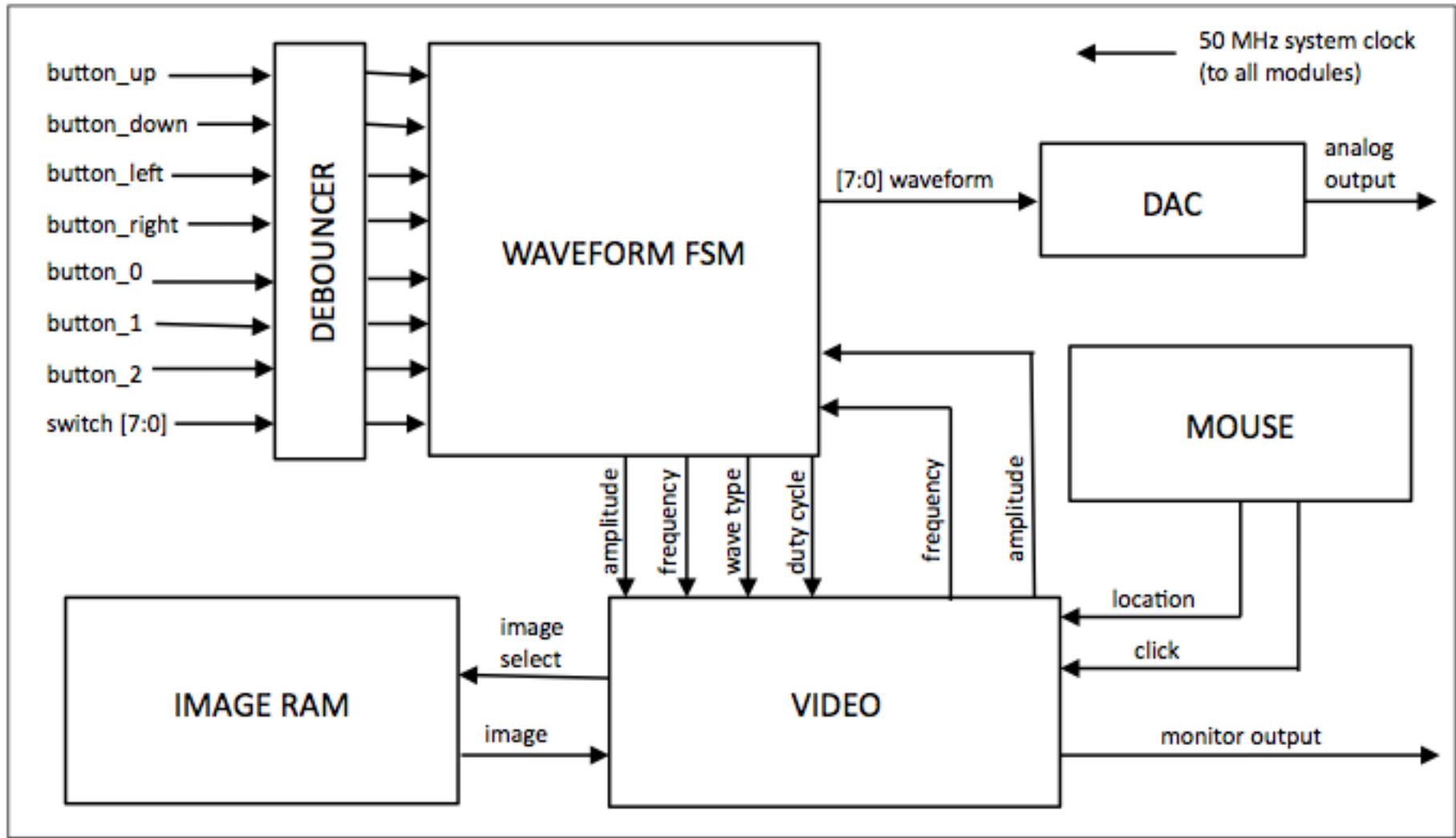
Ramp



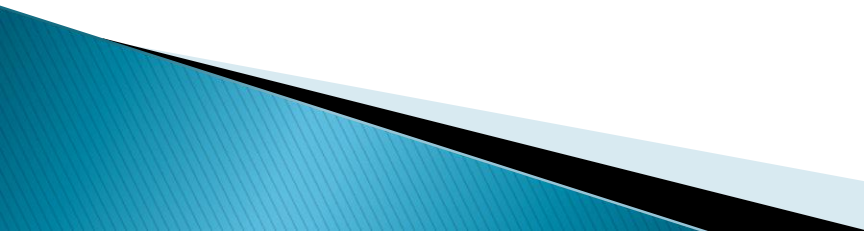
Triangle



# Block Diagram



# Waveform FSM

- ▶ Inputs: waveform type, freq, amp (buttons)
  - ▶ Outputs:
    - 8 bits to DAC
    - freq, amp, duty cycle (to display module)
  - ▶ 4 modes of operation (sub-FSMs)
    - Square: alternate between 2 values
    - Ramp: count up to a value, reset to 0
    - Triangle: count up, count back down
    - Sine: most difficult, using CoreGen sin function
- 

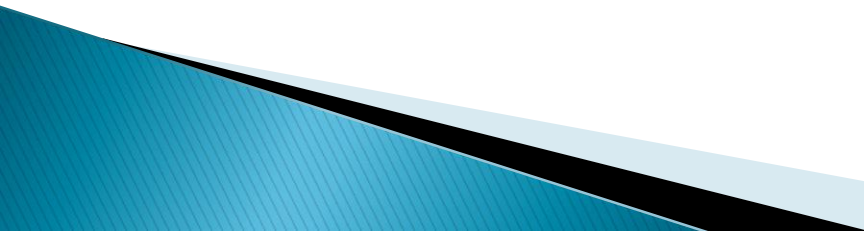
# CoreGen

- ▶ Uses LUT to calculate  $\sin(\Theta)$
- ▶ Stores half or quarter wave on dist. ROM
- ▶ Limited to 8 bits of output to DAC
- ▶ Using 10 bits of input for  $\Theta$  to make as smooth a curve as possible
- ▶ 80 core resource utilization...not much

# DAC

- ▶ 8 bits, Analog Devices AD7224
- ▶ Settling time of  $7 \mu\text{s}$   $\Rightarrow$  max freq 142 kHz
  - Our range: 100 Hz - 100 kHz
- ▶ Can swing from 0V to  $(255/256)V_{\text{REF}}$ 
  - Our range: 0V to 5V

# Video

- ▶ 50 MHz System Clock
  - ▶ Images from Image RAM
    - bitmap converted to RGB values in MATLAB
  - ▶ Real time images
    - uses output from Waveform FSM
  - ▶ Implementation of mouse control
- 

# Timeline

- ▶ Sunday 11/21
  - square, ramp, triangle waves functional
  - video module completely coded and tested using Image RAM
- ▶ Wednesday 11/24
  - sine wave functional
- ▶ Sunday 11/28
  - include real time images and basic mouse control
- ▶ Wednesday 12/1
  - integrate systems to make sure display and generation are synchronized
- ▶ Monday 12/6
  - testing complete and system functional, additional mouse control added
- ▶ Checkoff 12/7–12/9