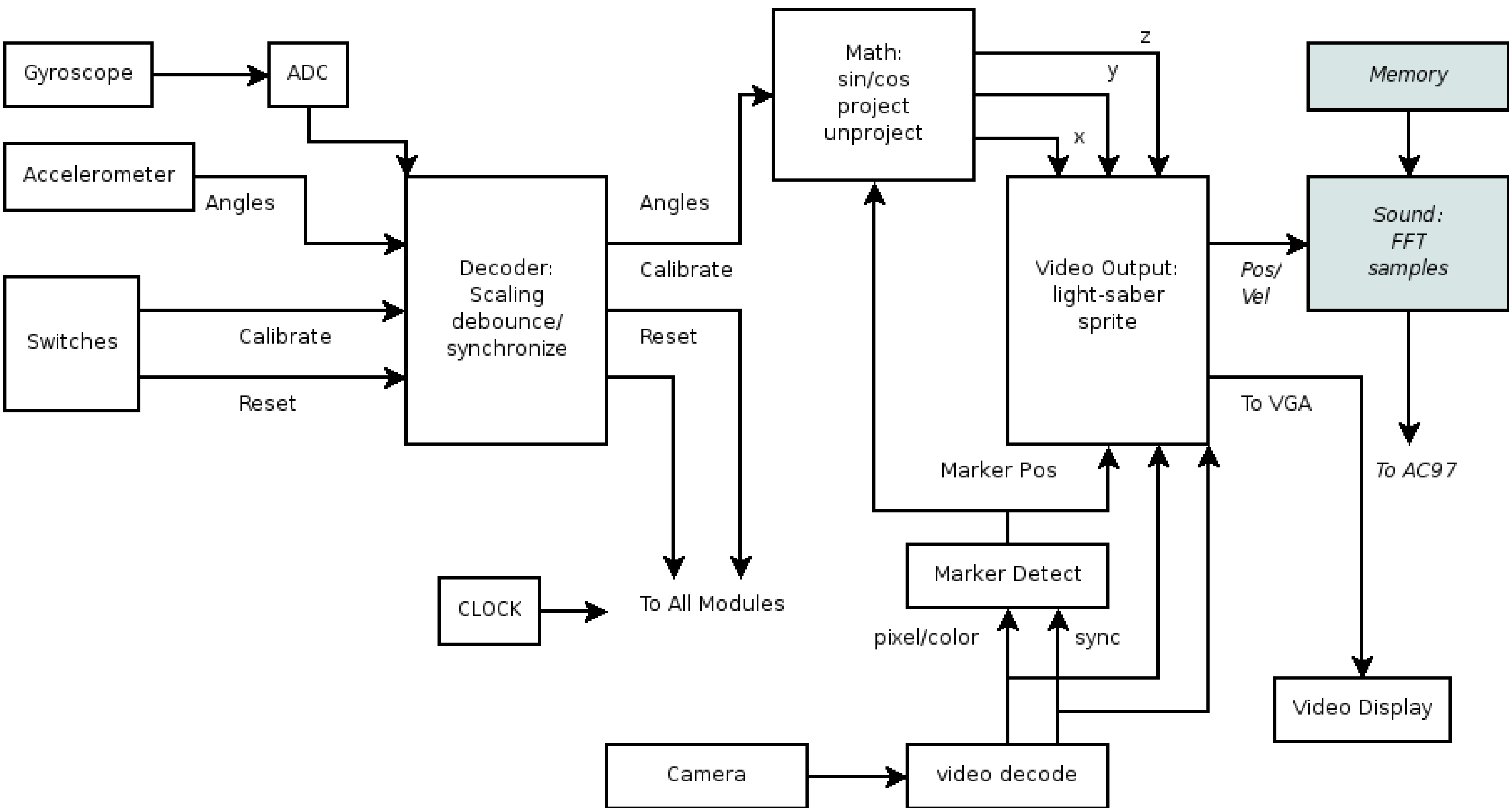
Real-time Lightsaber Generator

Joyce Chen, Michael Price, Hui Ying Wen

- Goals:
  - Generate live video of Star Wars lightsaber beam
  - Demonstrate realistic behavior in a duel
- Inputs:
  - Inertial sensors in lightsaber prop
  - NTSC video
- Output: VGA display
- Design challenges:
  - Obtaining accurate, uncorrupted sensor data
  - Tracking the lightsaber pose
  - Properly accounting for perspective in the shape of the lightsaber
- Cool, optional features:
  - Stereo sound with Doppler effect
  - Glow and motion blur
  - Multiple lightsabers

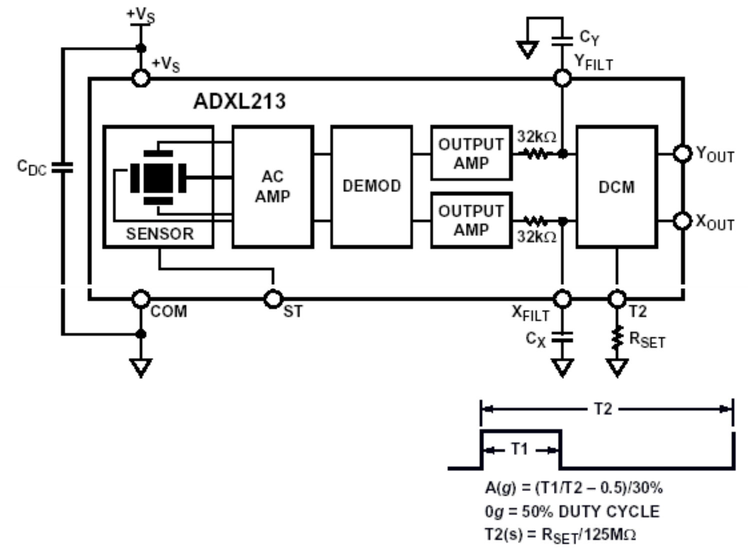
Block Diagram



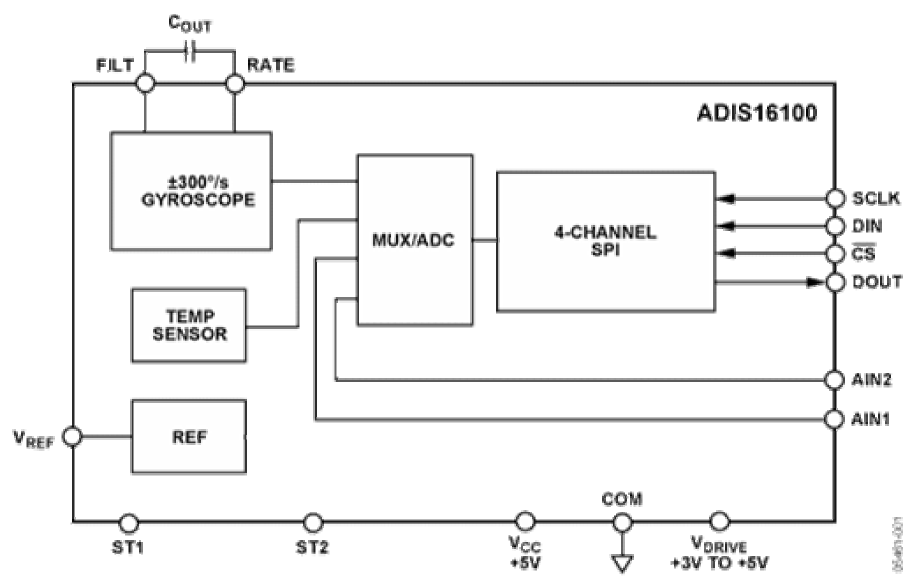
Inertial Sensors

Joyce Chen

- Accelerometer: Analog Devices ADXL213
  - Low Cost 1.2g Dual Axis Accelerometer Measure both dynamic and static acceleration
  - Use Cx and Cy capacitors to select bandwidth.
  - Output typically has bandwidth of 2.5 kHz



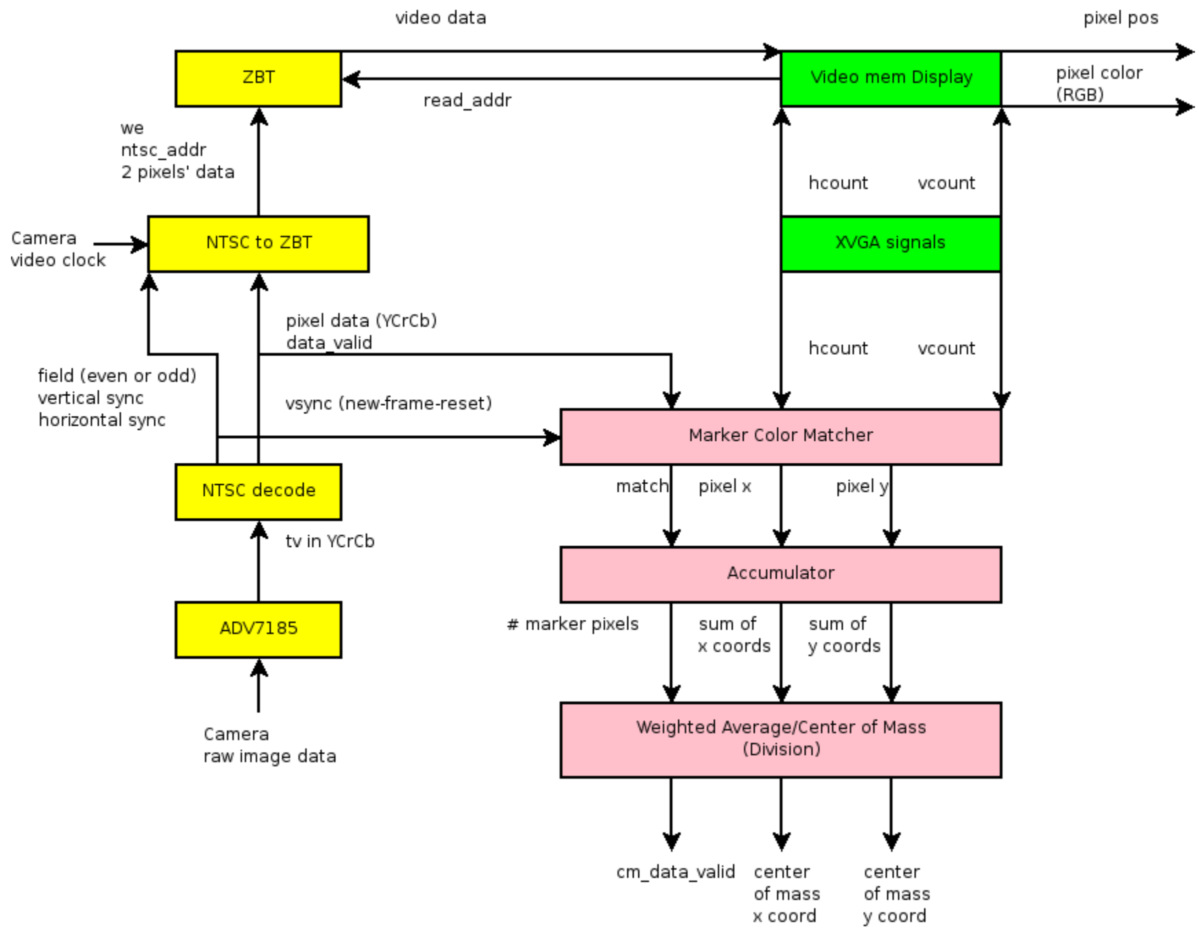
- Gyroscope: Analog Devices ADIS16100
  - ±300°/sec Yaw Rate Gyro with SPI Interface
  - z-axis rate detection : positive output voltage for clockwise rotation about axis.



Video Input & Marker Detection

Joyce Chen

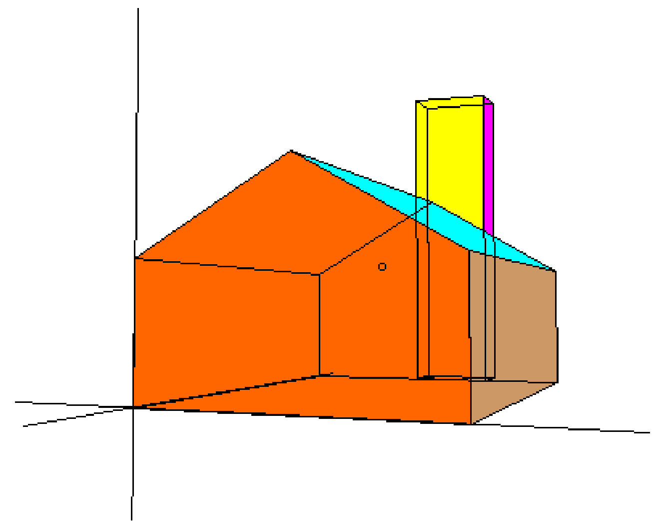
- Staff video module with ZBT video memory
- Marker detected by colour
- Position of marker filtered for noise and returned to Video Output module



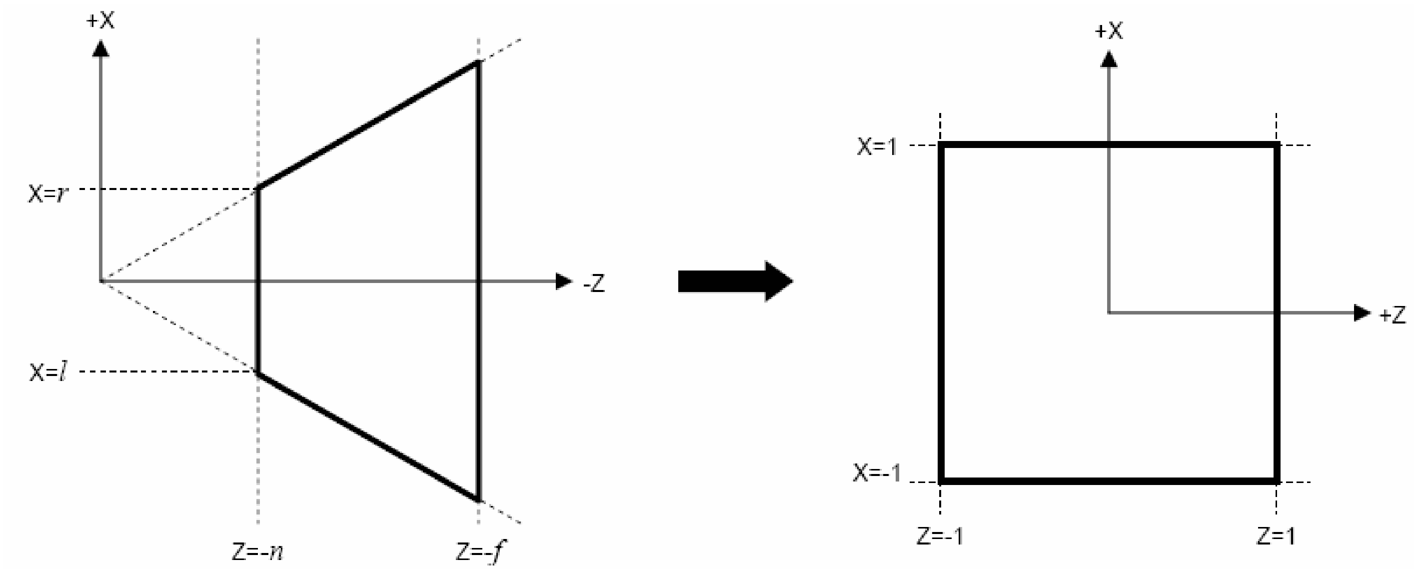
Perspective transformation

Michael Price

- A cylindrical lightsaber beam is:
  - Parallelogram if viewed in an orthographic projection
  - Trapezoidal in reality (each of 4 corners must be treated separately)



- Problems
  - Rotate and translate to match real position and orientation of lightsaber
  - Convert a field-of-view volume (global coordinates) into a flat rectangle (screen coordinates)



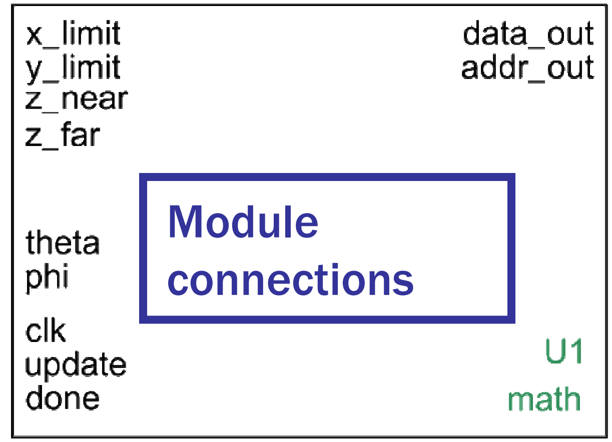
Math module

Michael Price

- Numerical format
  - 18 bit, fixed point, two's complement
  - Range: -8 to 8 meters
  - Resolution: 0.25 mm
  - Homogeneous coordinates [x, y, z, w]

- Parameters and inputs
  - Measured angles [phi, theta]
  - Measured position of marker on screen
  - Lightsaber coordinates (X_local): 4 points at corners
  - Boundaries of view volume (assume symmetry to reduce number of variables)

- Method: 3 phases – matrix storage in RAM
  - Generate matrices based on sensor input
    - Rotation/translation (R) and perspective projection (P)
  - Multiply: X_global = R * X_local
  - Multiply: X_screen = P * X_global



Block diagram (not shown)

Dependence on intermediate RAMs

Update signal triggers sequence of matrix calculations

Other modules access screen coordinates from memory

Video Output

Hui Ying Wen

Inputs

- hcount, vcount, hsync, vsync
- From Video Input module: saber_xbase [10:0], saber_ybase [9:0]
- From Math module: x and y values of 4 points of saber

Outputs

- pixel (RGB, to monitor)

Description

- Sprite module: draws light-saber on top of camera input
- Tests whether current hcount, vcount inputs fit within four boundaries of saber
- Complexity: multiplication and division on fixed-point values of 18-bit precision. No significant RAM memory.
- Also handles video output from ZBT memory, Y'CbCr to RGB conversion
- Extras: shading, blur

Timeline

Nov. 27 (after Thanksgiving Break)

- already have operational individual modules
- start interfacing between modules

Dec. 4

- have handle built and interfaced

Dec. 11

- presentation and report