Bryan Morrissey – Hardware Interface and Data Acquisition System

Non-Verilog Tasks:
- Output Pulse Amplifier
- Delta-Sigma Interface Circuitry (12 Channels)
- Ultrasonic Transducer Arrays: Transmit Array, Receive Array

Verilog Modules:
- FSM for interface control logic
  - output pulse trigger
  - subsystem timing
  - threshold control
  - ready/wait signals for other blocks
- Output Pulse Generator
  - optional variable timing, variable pulse width (number of 40kHz cycles per pulse)
- ADC and Data Acquisition Logic
  - Sync and Feedback latches
- Waveform reconstruction
  - converting 1-bit 27MHz signal into multi-bit 1MHz signal
  - Initial LPF and DC bias compensation
- IIR High Pass Filter – low-order filter for DC component blocking
- FIR Low Pass Filter – block 1MHz sampling/quantization/aliasing noise
- Zero crossing detector – convert multi-bit signal to one-bit signal
  - Optional feature – adjust threshold to compensate for stronger/weaker reflections
- BRAM Data Buffer – 12-bit x 32K
  - buffer 1-bit datastream from 12 receiver channels
  - store up to 32,768 samples at 1MHz sample rate (30Hz refresh rate: 33.3 ms/cycle)
  - one read port for signal analysis block
  - one write port (optional read/write to access raw data for debugging)
- Optional Components (If time allows)
  - RS-232 serial interface to access raw data
Alternate video display mode to visualize raw transducer data

**Zhen Li – Signal Analysis System**

Basic Functions:
- Wave Package Detector
  - Read bit stream from BRAM and determine the boundaries of reflected wave packages appropriately
  - Input: BRAM, start; output: (t1, t2) (can be shown on LED display), done

- Direction Sweeper
  - Read a single wave package and extract the angle information (inter-receiver delay dt) from it.
  - Input: BRAM, (t1, t2) from Wave Package Detector, start; output: dt (can be shown on LED display), done

- Coordinate Retriever
  - Calculate distance r and angle cos(θ) from t1, t2, dt; then convert the coordinates into (x,y)
  - Divider and square-root modules can be checked separately
  - Input: t1, t2, dt, start; output: (x,y) (can be shown on LED display), done

- Parameter Manager
  - Normal mode: provide coefficients (a1,b1,a2,b2)
  - Calibration mode: calibrate coefficients (a1,b1,a2,b2) from a set of (t1, t2, dt)’s
  - Input: t1, t2, dt, start; output: (a1,b1,a2,b2) (can be shown on LED display), done

- Controller
  - Signal each block to start, and receive done flag from each block
  - Communicate with the other two systems
  - Input: start signal from Hardware Interface and Data Acquisition System, a set of done's from each internal block; output: done signal for Display and User Interface System, and a set of start's for each internal block (can be shown on LED display)

Improvements (if time permits...):
• Width of objects
  o Direction Sweep extracts the width (tw) of the peak, and Coordinate Retriever estimates the width of the object from (t1, t2, dt, tw)
  o Calibration on the width calculation
  o It's a modification of Direction Sweep, Coordinate Retriever, and Parameter Manager.

• Multi-object Recognition
  o Retrieve more than one object from one wave package, i.e. at the same distance r
  o It's a modification of Direction Sweeper, Coordinate Retriever

**Brian Wong – Display and User Interface System**

Data processing
• Double-buffering (if necessary)
  o Use ZBT or BRAM to double-buffer XVGA resolution screen output

• Motion detection module (if time permits)
  o Ability to filter out incoming noise to produce a stable object location
  o Use memory to record object location history
  o Use location history to produce object average speed estimation every second

• Object representation module
  o Represent object with different speed and color
  o Ability to track object and hold consistent coloring for each object in range
  o Change object size or color based on distance from sensor or object movement

User interface
• Sonar scope layout module
  o Create Java program to convert sonar scope layout into .COE bitmap
  o Initialize data into labkit memory unit (ZBT or BRAM)
  o Draw sonar layout as background on XVGA screen
● Text display module
  ○ Display angle and distance estimation data for each object
  ○ Display speed estimation data for speed vector
  ○ Display basic system setup such as scan rate and audio pulse frequency on screen
  ○ Able to float text to the side of each object

● Vector generation module (if time permits)
  ○ Take object movement estimation and draw vector
  ○ Change proportional vector size based on the speed magnitude
  ○ Locate object vector intelligently next to object in range

● Sound generation module (if time permits)
  ○ Generate sonar detection indicator sound if objects are detected in range
  ○ Generate warning siren if object is too close to sensor or is moving in certain direction or speed exceeds threshold

● Sweeping sonar line module
  ○ Generate a sonar scope sweeping line that rotates around the scope center
  ○ Create an alpha-bending effect such that there is a trailing edge on screen
  ○ Interact with sound generation module to produce a sonar “ping” when line intersects with an object on screen (if time permits)

Testing and Debugging

● Calibration module
  ○ Provide a simple interface for calibration sonar sensor
  ○ Display calibration data on screen

Improvements

● Motion recording module (if time permits)
  ○ Record object location in memory
  ○ Provide basic playback mode for to display object movement on screen