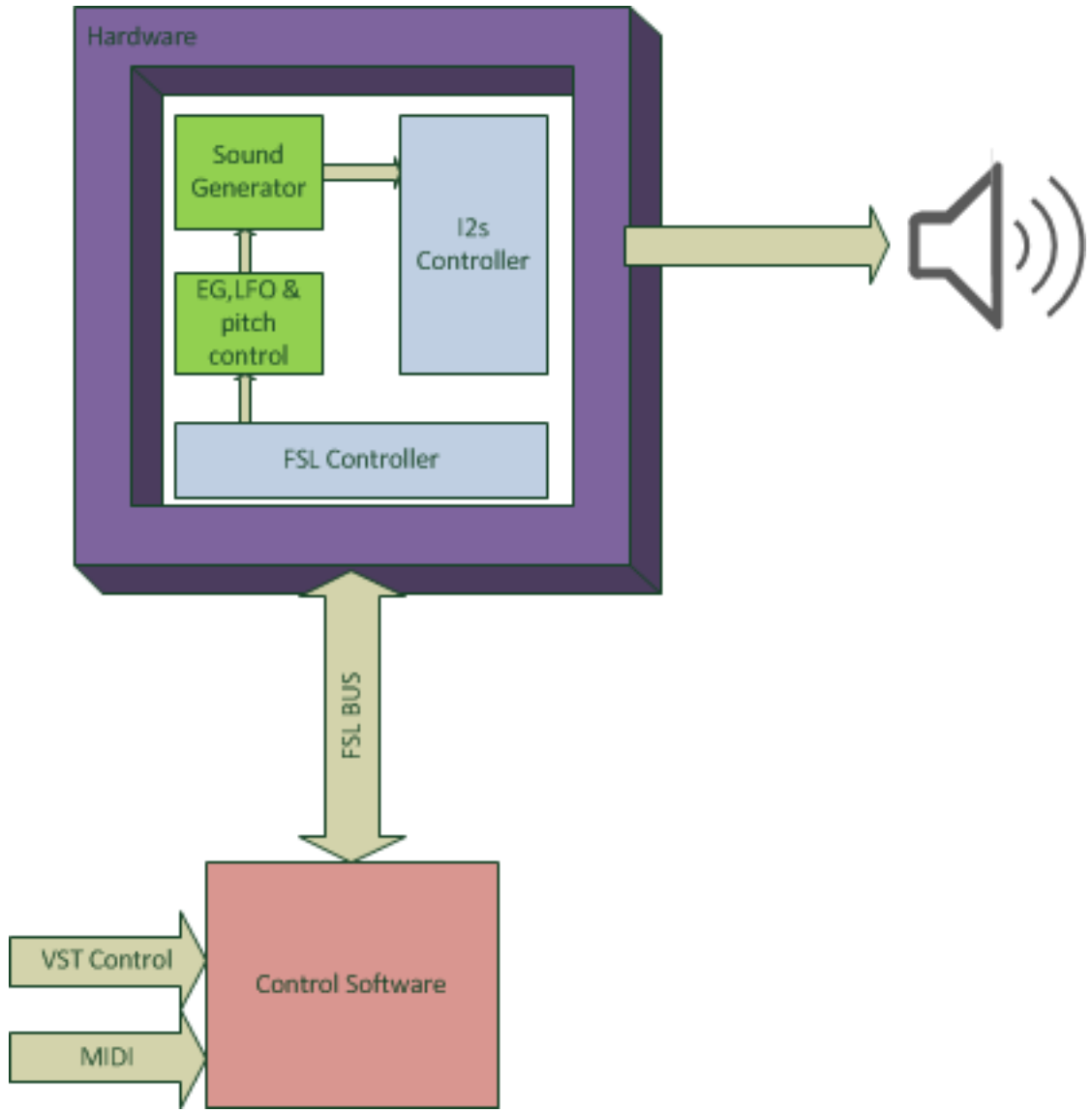
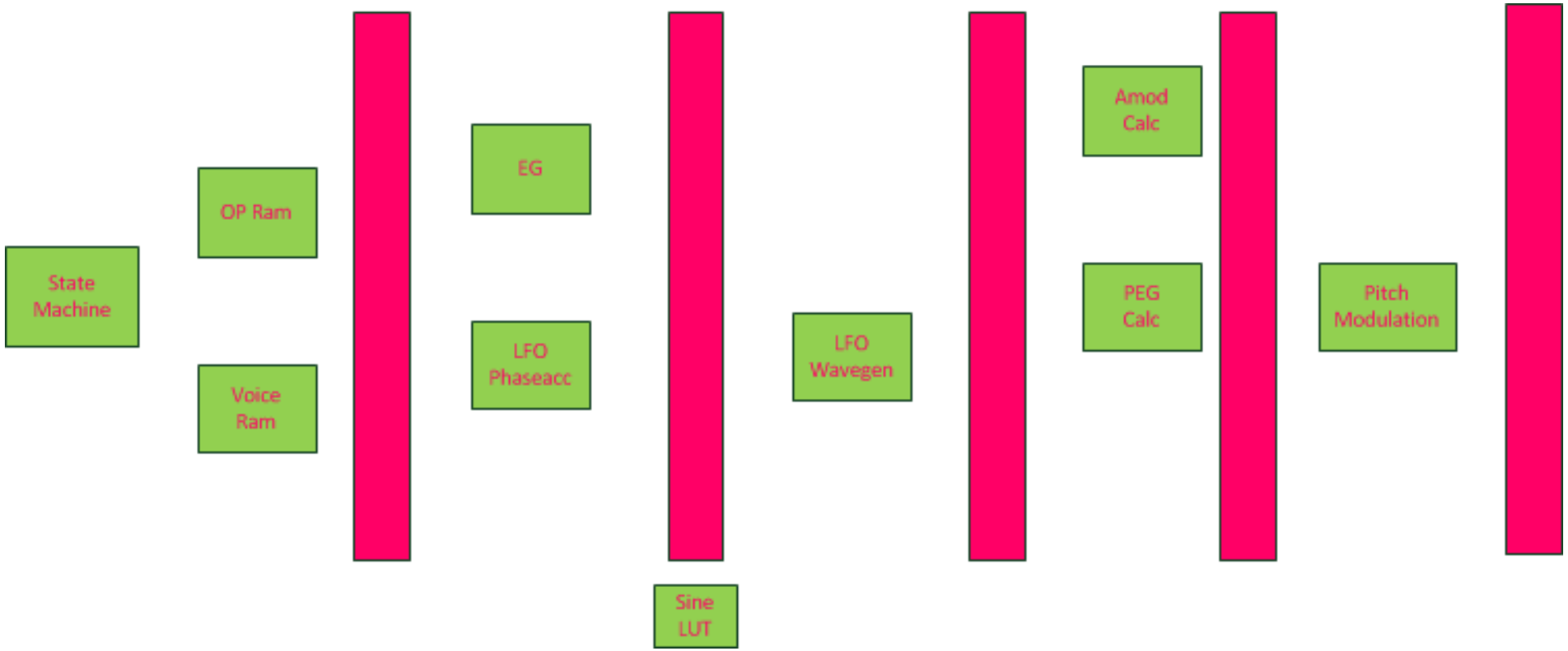
FPGA - DX7

Andreas Irestål
(dt07ac9@student.lth.se)
Rajanarayana Priyanka, Marigi
(aso10rma@student.lu.se)
Shadi Abdu-Rahman
(dt07sa6@student.lth.se)

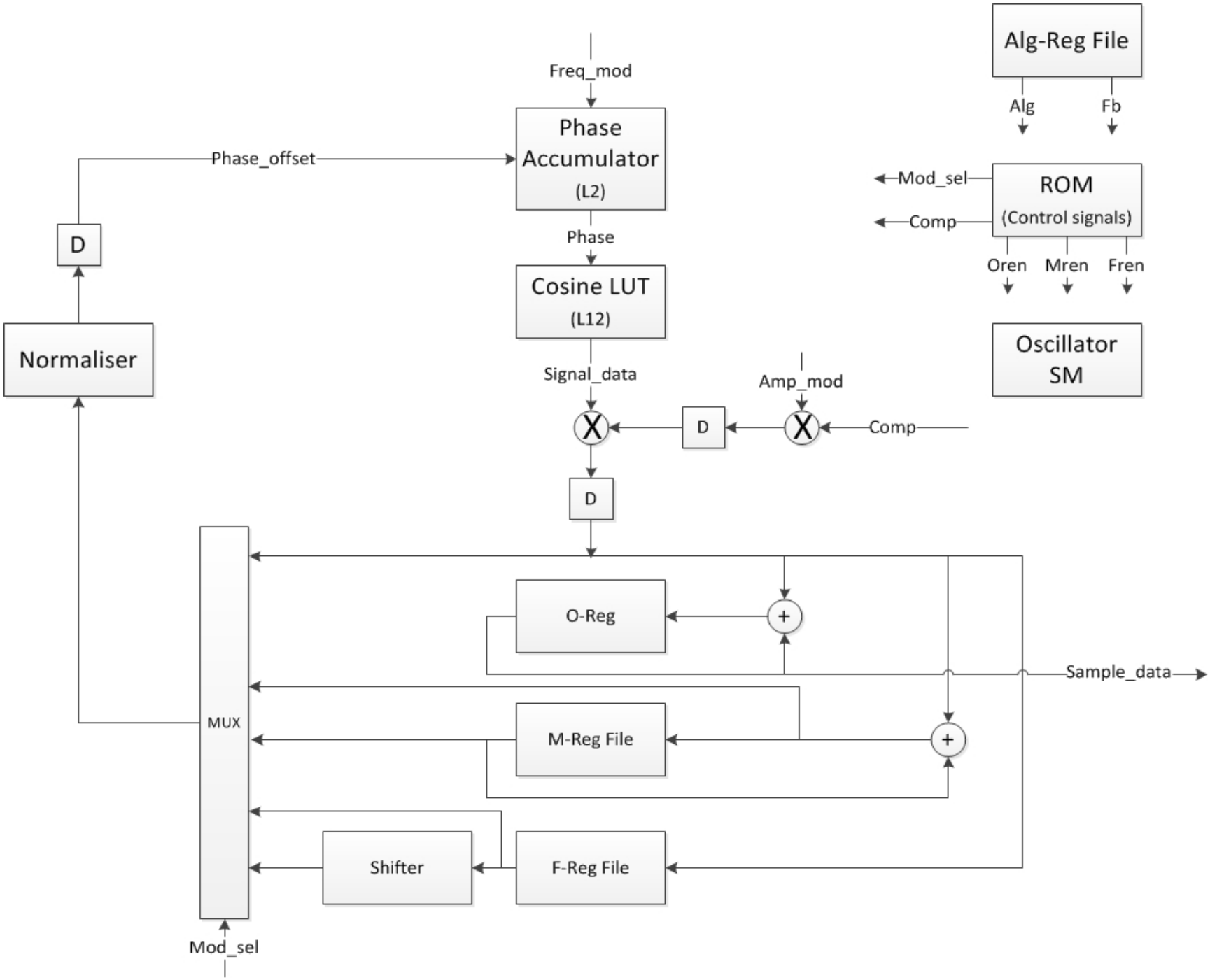
Hardware/Software Architecture



FPGA HW: Envelope Generator, Pitch Envelope Generator & LFO



FPGA HW: Sound Generator



FPGA HW: FSL Controller, Mixer & I2S Controller

- FSL controller forwards the data message to the correct recipient (EGS, OPS, Mixer)
- Mixer controls the volume and panning parameters of the output sound (stereo)
- I2S controller serializes the sample data and sends it to an I2S DAC Pmod
- Approx. 52.08 KHz sampling frequency, 24 bit depth.

MicroBlaze SW

- Custom Pmod with USB-RS232 & MIDI ports for communication with MIDI keyboards and VST Plugin
- Interrupt driven USB & MIDI UART communication (1.5 MBaud and 31,250 Baud)
- Decodes MIDI Messages (Notes, Pitch Wheel, Sysex Patch Changes)
- Decodes USB messages (Notes & Parameter Changes)
- Sends all decoded messages to synthesizer hardware
- Converts DX7 parameters into our more convenient format to simplify hardware

VST Plugin

- For use in music production software so that changes to synthesizer parameters and recorded melodies can be stored and sequenced from software

- Uses USB to send messages to the FPGA

Features Implemented

- Sequencer (VST plug-in) & MIDI keyboard interface support
- Polyphonic / Monophonic voicing
- Real time patch loading
- Higher resolution parameters than a conventional DX7

Unimplemented Features / Possibility To Extend Features

- Polyphony of 64 voices possible (provided more memory)
- LFO have not been measured and implemented in the same format as in a DX7
- Physical parameter interface (knobs, switches, LCD)
- Patch Bank stored on FPGA with physical preset browser hardware (switches, LCD)
- As of now, the VST Plugin can only be instantiated on one channel. It could be extended to several instances controlling different voices of the FPGA Synthesizer
- The audio output from FPGA synthesizer could be sent back to the VST plugin via USB for recording in music software
- Multitimbrality (ability to play on different midi channels at the same time, a.k.a piano and organ at the same time) could be implemented by doing a small extension to SW. It's already there in the data structures.

Problems Encountered

- Limited hardware resources especially Multipliers and Block RAMs which are extensively used
- Initially obtained very low operating frequency
- The controlling hardware was very big and hard to implement. 5 Pipeline stages were needed
- Noise issues due to low amplitude resolution. Solved by increasing and interpolating
- Exponential representation of parameters.

Solutions

- To solve limited multipliers issue panning was disabled in hardware
- Adapted hardware to minimize critical paths to obtain operating frequency
- Exponential Parameter values were either bit shifted, tabularized, or interpolated linearly

Lessons Learned

- Designing is an iterative process
- User friendly features are not necessarily engineer friendly!
- Steinerg's VST SDK is messy & poorly documented
- The MicroBlaze interrupt system is well thought-out but poorly documented

Conclusion

- This project may have been too big for only 7.5 credits and 3 persons
- DX7 have very inconvenient representations
- FM Synthesizers still rocks
- Always assume you're wrong, not the hardware