Bass Active Crossover Filter: Final Presentation

Team M4:
Albert Hwang
John Kuhns
Adrian Loh
Andrew Ryan

Overall Project Objective:
Design a 16-bit low-/high-pass filter with three selectable cut-off frequencies

Stage: 23 Apr, 1997
Final Presentation

Introduction and Applications

- 16-bit low-/high-pass DSP low rolloff-frequency filter
- Three selectable cut-off frequencies:
  80 Hz, 120 Hz, 160 Hz
- Serial input/output (not enough pins for 16 in/16 out)
- Applications:
  - Hi-Fi car stereos
  - Home theatre systems

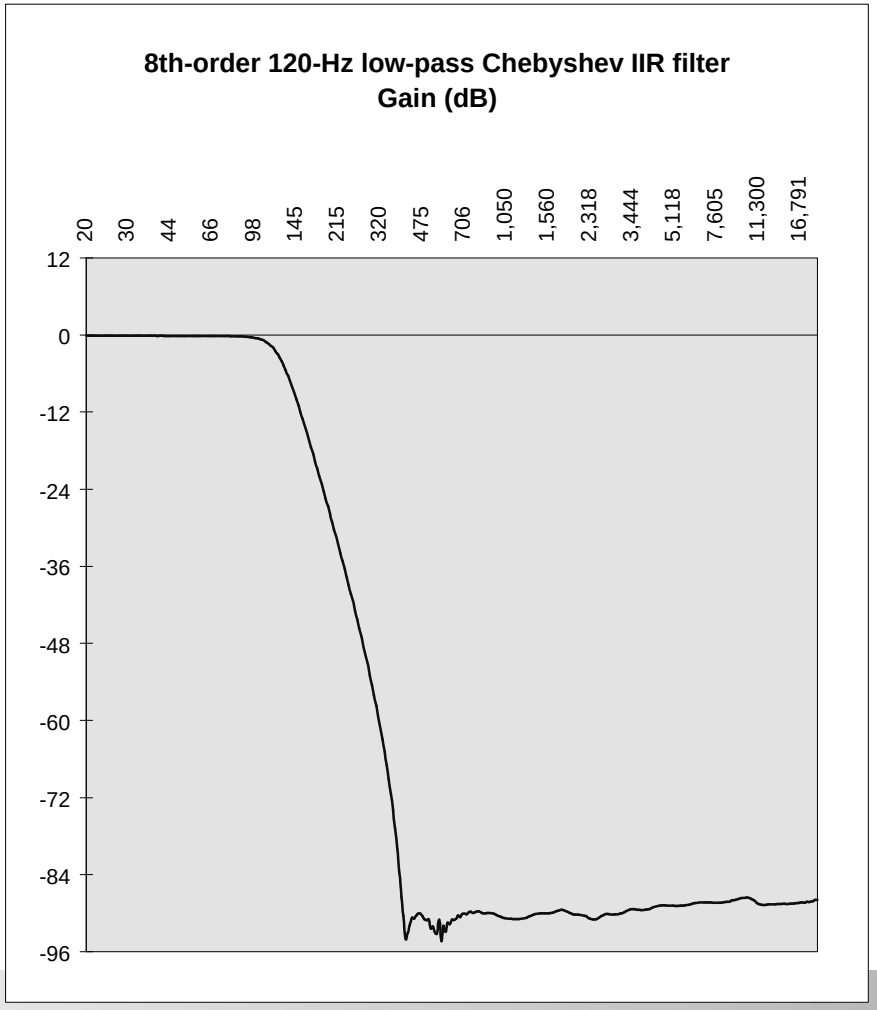
The Filter Algorithm

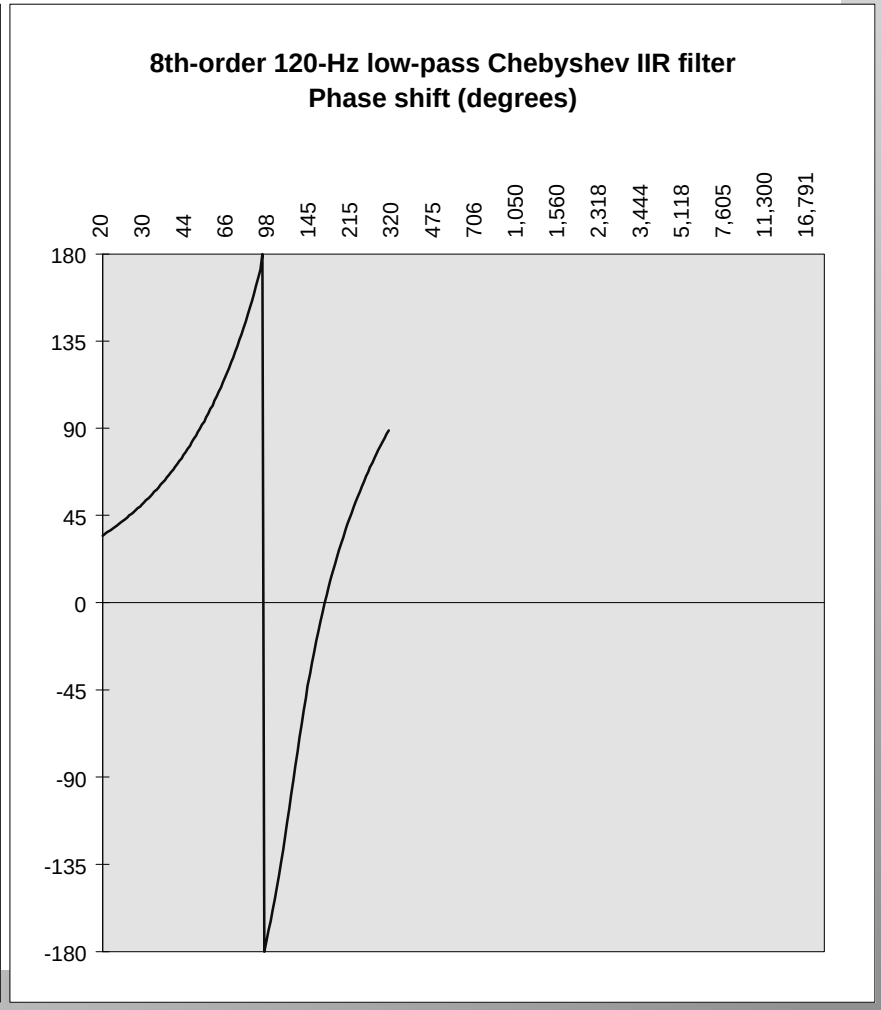
- Chebyshev Type II filter
- Minimized passband ring error
- Four cascaded 2nd-order filter sections: 8th order filter
- 20-bit constant coefficients
- 32-bit datapath and SRAM state variables, for values in intermediate calculations
- Approximately 700,000 32-bit MACs per second

Performance

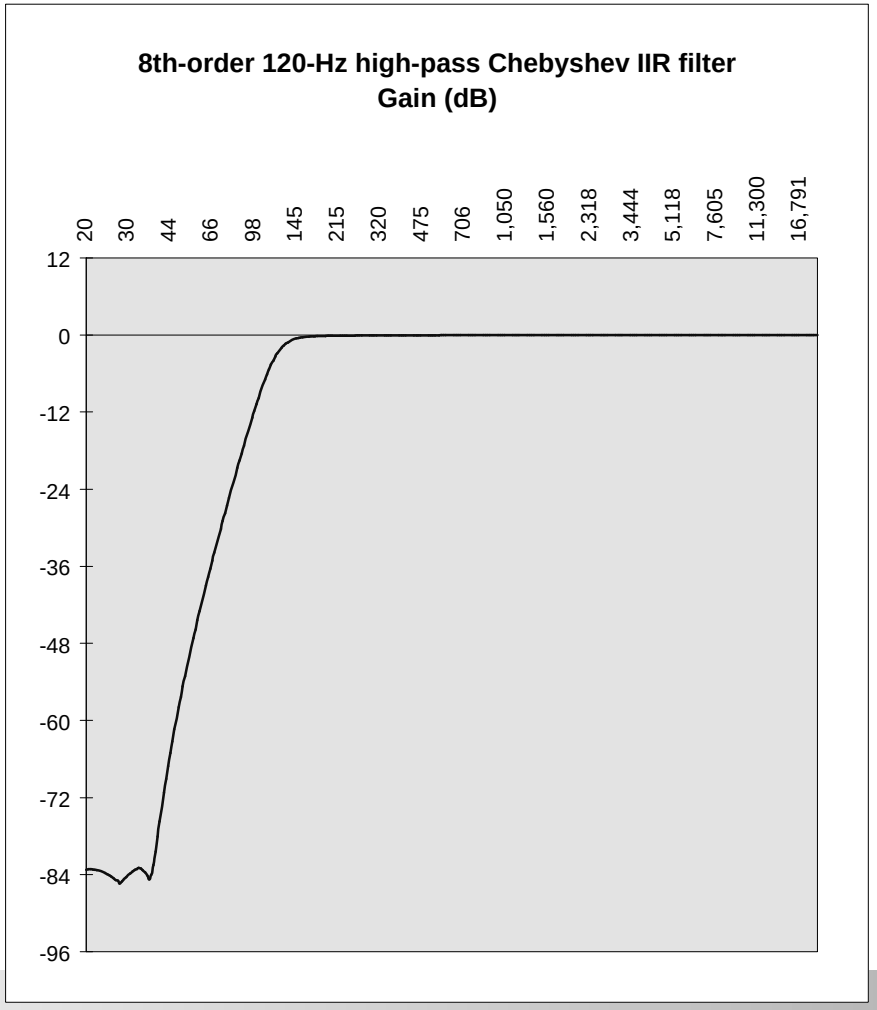
- 44.1 to 48 kHz sampling frequency
- 16-bit input/output
- 48 dB/decade rolloff
- >80 dB stop-band attenuation
- ±0.5 dB pass-band ring error
- ±0.5 dB combined low-/high-pass gain error
- 16 MHz system clock

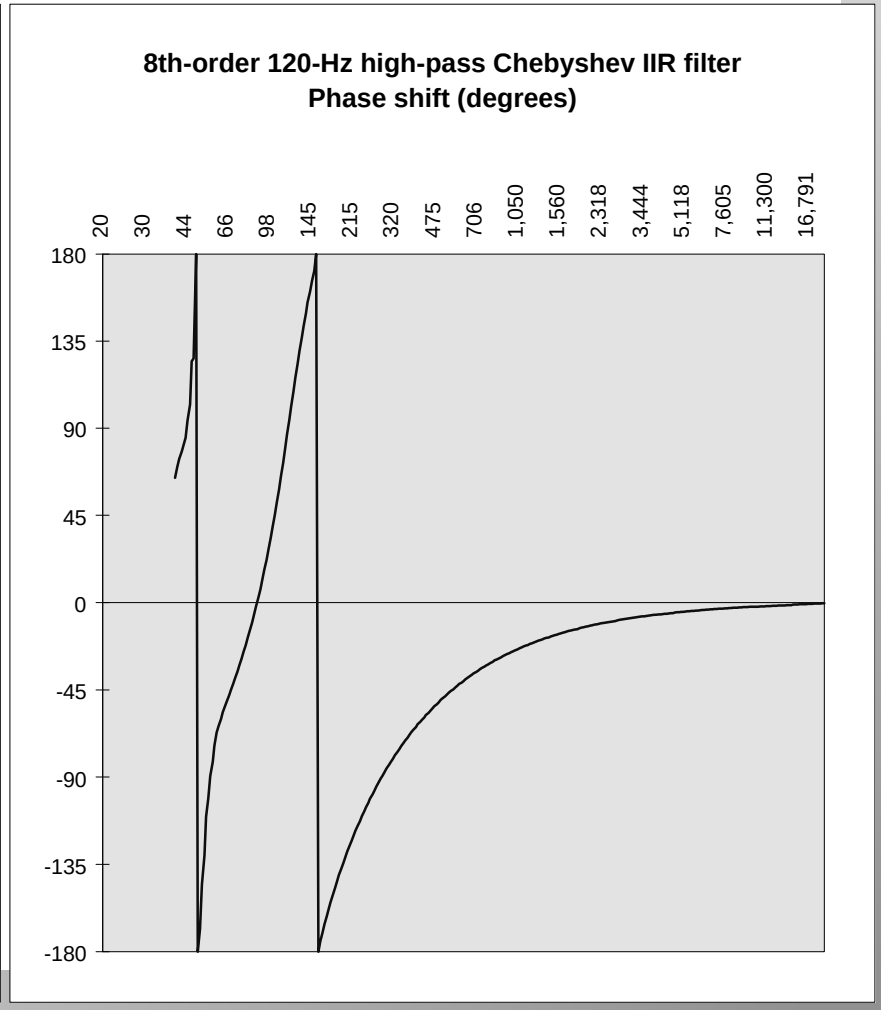
120-Hz Low-Pass Filter



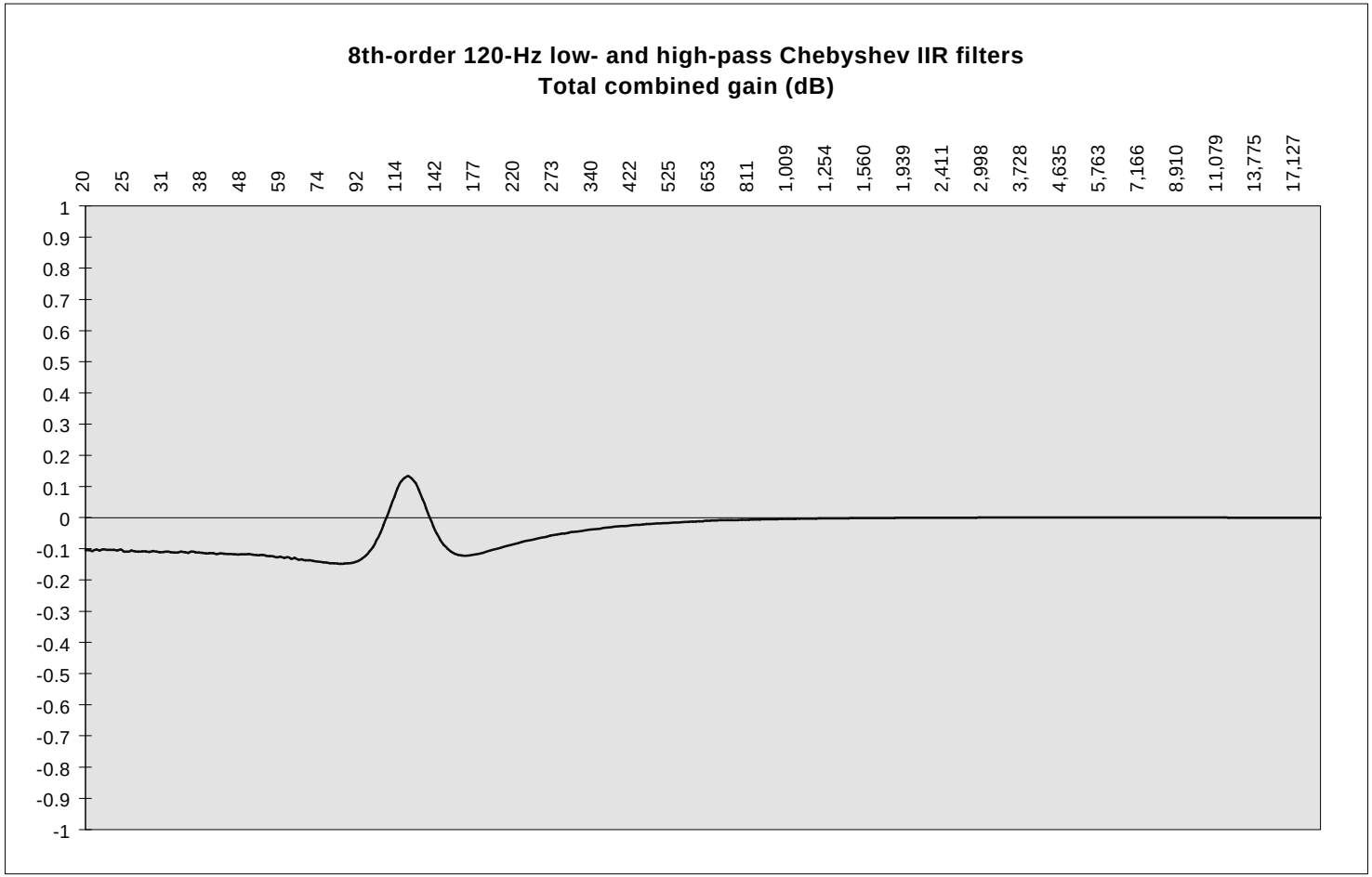


120-Hz High-Pass Filter





120-Hz Filters' Combined Gain



I/O Timing

- Three clocks:
  - Word “clock” (~48 kHz) -- Each 16-bit sample
  - Bit clock (~724 kHz) -- Each serial bit in/out
  - System clock (~16 MHz) -- Each internal operation
- Bit clock is exactly 16 times faster than the word clock
- The system clock is decoupled from the other two clocks
- The word clock resets the program counter and starts a new DSP sample computation (consisting of 15 MACs).
- The word clock is sychronized with the system clock
  - Uses multiple DFFs and inverters
  - Nearly eliminates metastability glitches between word clock and the system clock.

Datapath

- 32-bit internal datapath
- 16-bit external input/output
- Computes multiplies in sign and magnitude format, but additions in two's compliment (must switch on the fly)
- Made up entirely of DFFs, muxes, and a 32-bit adder
- Adder made up of eight 4-bit carry-lookahead units

Control

- Originally over 300 states with 16 control bits/state
- Reduced to 52 states with 9 control bits/state using run-length encoding and eliminating mux select redundancy
- Uses State ROM to store the “program”
- “Glue logic” controls
    - 6-bit “program counter” for filter operations
    - 5-bit “loop counter” for state repetitions (i.e. multiplies)

Floorplan

- Initial estimates: 14,240 transistors
- Final total transistor statistics:
    - 10,310 transistors
    - < 320 λ^2/transistor
- Use of M2 as “channels” in datapath

Channels

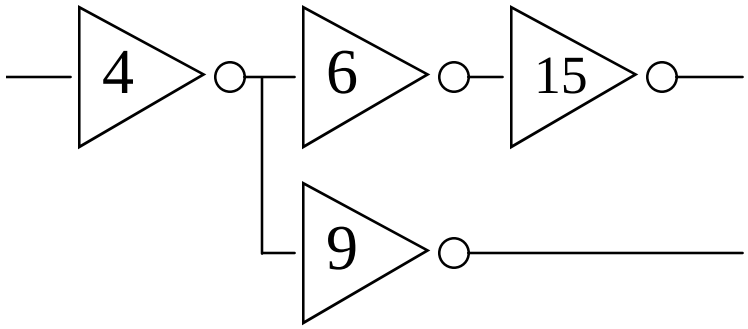
- Extensive use of M2 (channels) for routing in datapath
- Routes GND lines and interconnect between datapath blocks
- Eliminates wasteful 32-bit corner routing
- 5 channels/bit * 8 λ/channel * 32 bits x 1280 λ

Timing

- Critical path is SRAM reading/writting.
- SRAM null state -- SRAM “double clocked:”
  - On clock high, SRAM addresses setting up, but write enable forced off
  - On clock low, SRAM addresses slowly de-charge, while write enable is allowed to quickly charge up
  - Eliminates false write glitches in IRSIM and HSPICE
- Timing for other blocks (must be below 62.5 ns)
  - ROM (40 ns) + 2 gate delays = 45 ns
  - Muxes (14 ns) + Adder (37 ns) + DFFs (3 ns) = 54 ns

Buffers

- Buffers are important to boost strength of mux selects on critical paths
  - Wrote C program to compute optimum buffer sizing
  - Produces balanced-delay complement signal
  - Buffer stages used on chip:
    - 1-2-2 (low-drive), 2-4-6 (medium-drive), & 4-6-9-15 (high-drive)
  - Clock driver:
    - 8-70 staged
    - 120 DFF (or 360 equivalent inverter gate) load
    - Oscillates between 1 and 4 V (works great)



Testing

- Wrote C program to implement DSP algorithm
- Modified C program to reflect our chip's precisions:
  - 32-bit SRAM filter states
  - 20-bit ROM filter coefficients
- Wrote Verilog to exactly match C
- Wired chip together to exactly match Verilog (and C)
- Testing samples in HSPICE unmanageable -- 300 cycles per sample!
- Tested chip using IRSIM, hacked two buffers onto chip.
- Got it working at 1am this morning, and the SRAM reset working at 7am this morning (whew!)

Conclusion

- It works!
- High-quality, high-fidelity DSP for real-world applications
- So how’s it sound? C, Verilog, IRSIM, fab(?) -- all the same