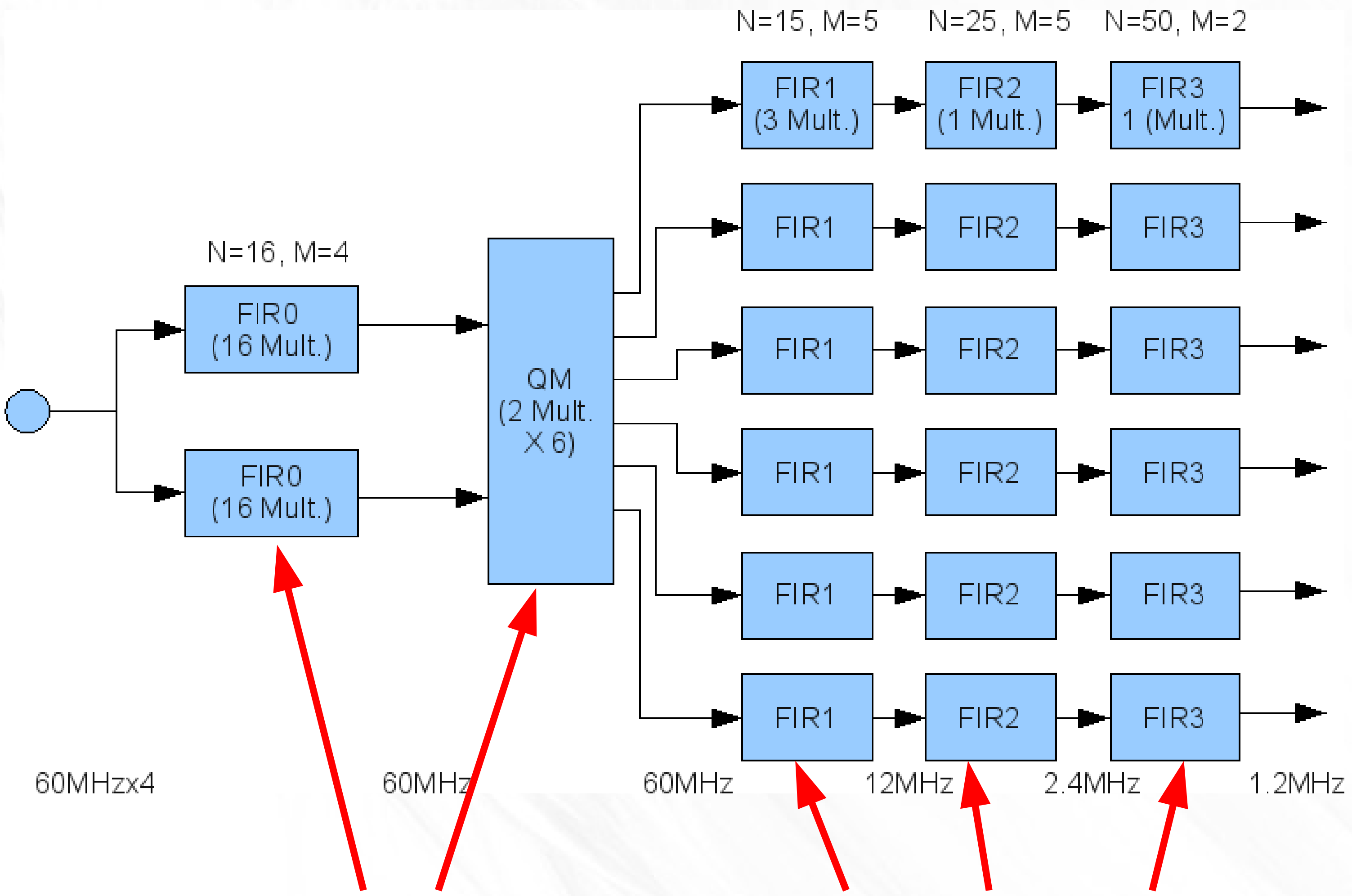
240MHz Digital Filter Tuning

Jason Zheng, Kayla Nguyen,
Charles Le, Yutao He

Overview



Need Tuning

Fixed Center Frequency

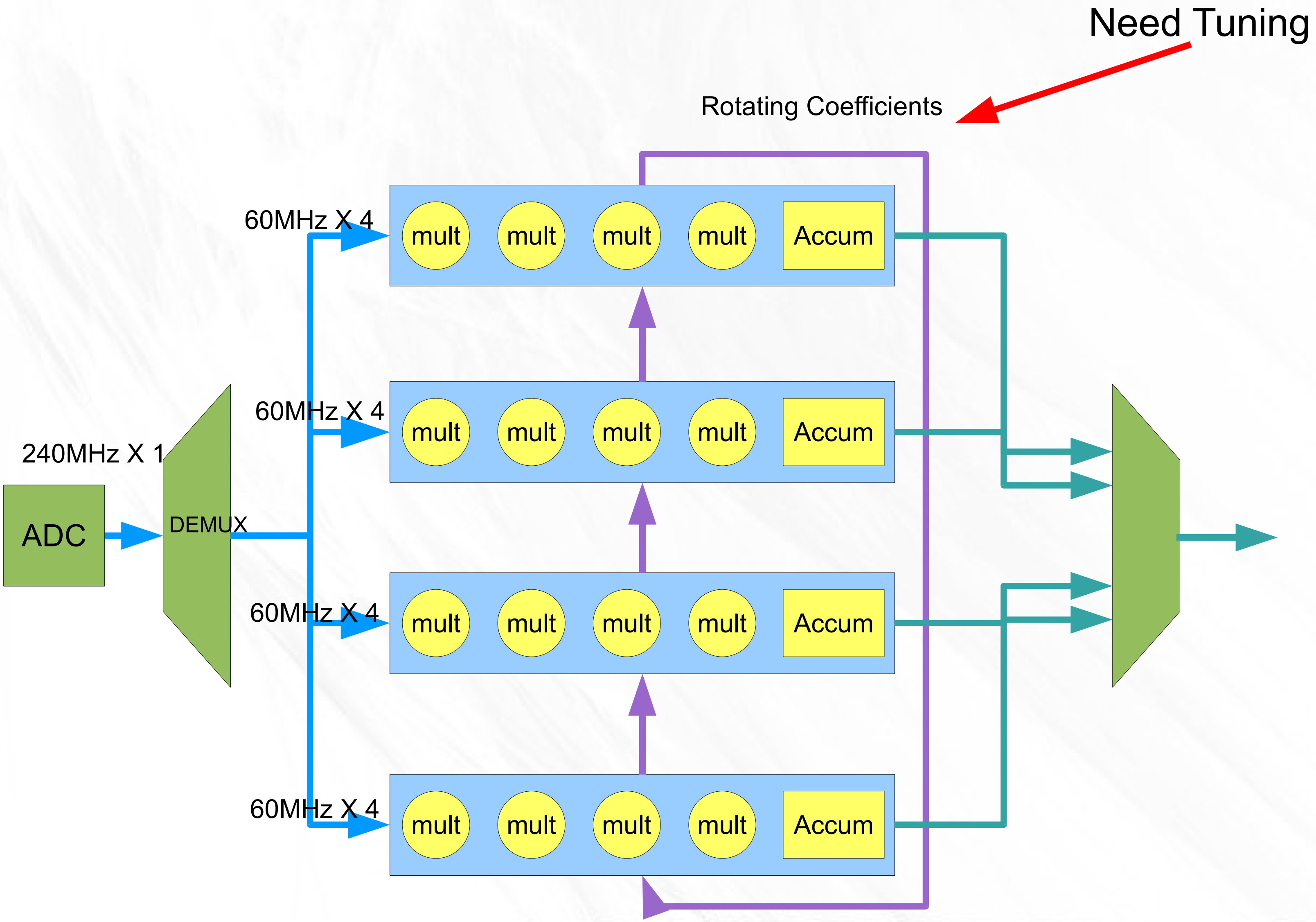
Overview, continued

- Assumptions:
  - Center (noise) frequency can move around in 20-80MHz range with 1MHz increments
  - Co-pol and X-pol frequencies are 2.5MHz around Fc
- How fast can we tune the digital filter?
- How much logic and storage overhead?

Cost Model

- Time to write a single coefficient/sine/cosine value to the FPGA T_{ld} = 1 clock
- Time to look up a value T_{lut} = 1 clock
- Time for a multiplication T_m = 1 clock
- Time for a sine/cosine computation T_{trig} = 40 clocks

Tuning FIR0


Need Tuning
Rotating Coefficients
240MHz X 1
ADC
DEMUX
60MHz X 4
mult
mult
mult
mult
Accum
60MHz X 4
mult
mult
mult
mult
Accum
60MHz X 4
mult
mult
mult
mult
Accum
60MHz X 4
mult
mult
mult
mult
Accum

Tuning FIR0

- Need to load the coefficient banks (32 loads)
- Need to computer the coefficients
  - wbpc= 2*pi*f_chanc/fs0;
  - hmsbpr= hmsbp.*cos(wbpc*[0:1:hlngbp-1]);
  - hmsbpi= hmsbp.*sin(wbpc*[0:1:hlngbp-1]);
- Option 1 – compute cosine and sine on the fly
- Option 2 – look up cosine and sine

Tuning FIR0 – Option 1

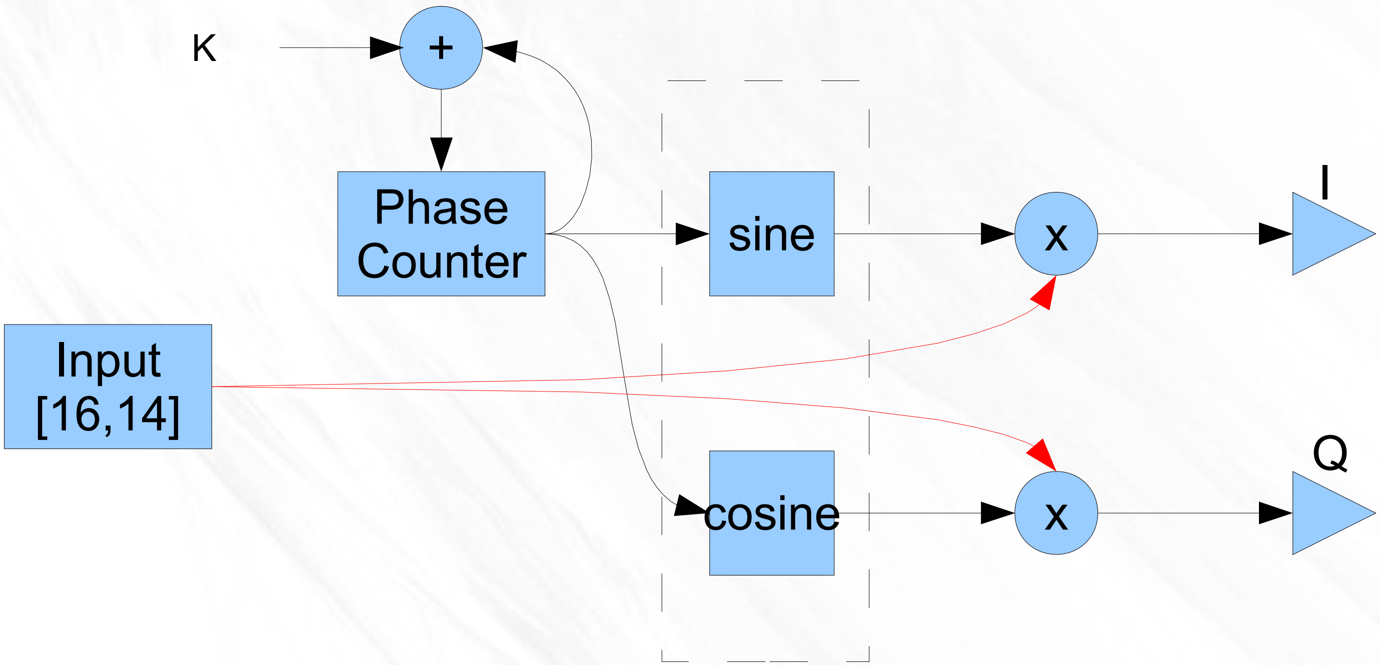
- Need a CORDIC core to compute sine and cosine
- Need 2 full multipliers (phase and final step)
- With pipelining, compute time is dominated by CORDIC: $T = 2T_m + 16T_{TRIG}$
- Total time for FIR0 = 32 + 2 + 16*40 = 674 clocks
- Fixed point errors = phase + CORDIC

Tuning FIR0 – Option 2

- Need a lookup table
  - 1MHz Center Freq. Incr. = 240 Entries (1.5º incr.)
  - 2MHz Center Freq. Incr. = 120 Entries (3º incr.)
- Need only 1 full multiplier for last step
- Need to look up 16 values (16 T_{lut})
- Total time for FIR0 = 32 + 1 + 16 = 49 clocks
- Phase errors is eliminated b/c LUT stores sine/cosine of the exact phase values

QM w/ Phase Counting

```
phase=wbpc*(0:N0-1)=2pi*fc/f0*(0:N0-1)
xbsr=xov.*cos(phase)
xbsi=xov.*sin(phase)
```



Phase count = phase*K
K = f0/(2pi)

Tuning QM

- Need to update each of the three the sine/cosine tables of size:
  - Need 480 entries (co-pol and x-pol has 2.5MHz diff.)
- Computing Sine/Cosine Values:
  - xbsr= xbpr.*sin(wshift*(0:Nbp-1)) - xbpi.*cos(wshift*(0:Nbp-1));
  - xbsi= xbpr.*cos(wshift*(0:Nbp-1)) + xbpi.*sin(wshift*(0:Nbp-1));
- Option 1 – recompute a partial table
- Option 2 – use preloaded full tables

Tuning QM – Option 1

- Need a CORDIC core and a full multiplier to compute the phase
- At 1MHz increment, total time = 480*40 + 1 + 480 = 19681 clocks
- At 2MHz increment, total time = 240*40 + 1 + 240 = 9841 clocks
- Fixed point errors = phase + CORDIC

Tuning QM – Option 2

- QM Tables do not need to be reloaded
- Only need to change a few registers (step size for the phase counters) to accomplish tuning
- Only need a nonvolatile storage to store the full table (free as part of configuration bitstream)
- Full tables can also be used for FIR0 tuning
- Phase precision errors are again eliminated

Scenarios

- Option 1 for both QM and FIR0 (1MHz):
  - Need at least 1 CORDIC core + 2 multipliers
  - No nonvolatile storage
  - 20354 clocks to finish tuning
- Option 2 for both QM and FIR0 (1MHz):
  - Need 1 multiplier
  - 480 entries of storage (free as part of bitstream)
  - Less than 60 clock cycles to finish tuning