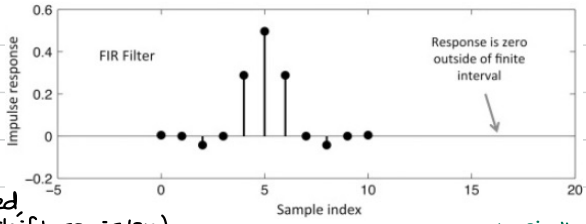
EE 287 PROJECT NOTES

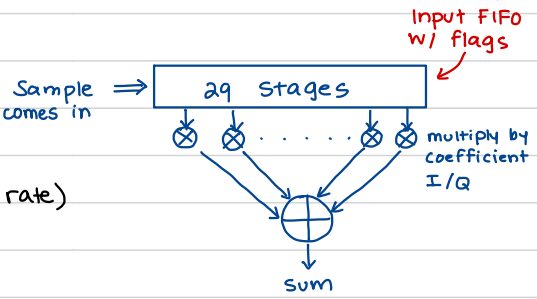
- a filter whose impulse response is of finite duration
- operate only on current and past input values
- provide a linear phase response
- always stable because they are implemented using an all zero transfer function

- Project: Finite Impulse Response (FIR) Filter
  - Consists:
    - 29 samples
    - multiply each by a constant
    - Sum together to get result
    - Everything is signed

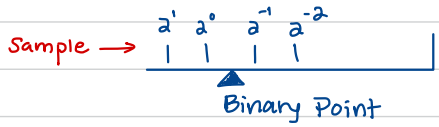


- First & last 14 coefficients are the same
- FIR contains last 29 samples for sampling
- When new samples are processed the oldest are discarded (like shift register)

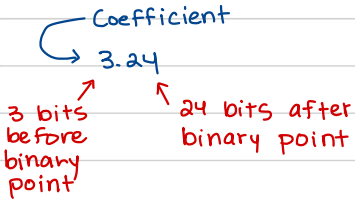
aka pipeline



  - Requirements:
    - Needs to run at 300 MHz
    - Limited to 5 complex multipliers (4 multipliers per complex multiply)
    - No more than 3 clock cycles per filter operation (100 MHz sample rate)
    - Input FIFO w/ two flag push model
      - Atleast 4 samples
    - Input data: fixed binary point 1.23 signed data (Range: $-1 < x < 1$)
    - Coefficients: fixed signed binary point 3.24 (Range: $-4 < x < 4$)
    - Output data: fixed binary point 8.24
      - Filter can have a gain >1
      - Higher precision is kept internally
      - Results are rounded towards zero (truncate positive values, increment negative values in position 8.25)
      - All values reset to 0 (including FIFO)



  - Notes:
    - TB always takes data (flags)
    - For bit size, add 5 bits extra above and below to both sides of binary point to avoid overflow (For Summation)
      - Adding 5 bits above/below will require summation to be 9.29 (38 bits)
    - Use DesignWare for multipliers
    - Capitalization matters (double check signals)
    - Synthesis will take approx. 4-5 hours (big design)
    - 5 complex multiplications each clock cycle
    - I is Real and Q is Imaginary
    - Data after a multiply of 3.24 and 1.24 is 4.47 format (only 8.24 will be in final answer)



  - Will provide the prompt on CANVAS
  - He will give TB this weekend (encrypted) for us to see how it works

Note:

The design can be simplified if the first and last data are added, and then multiplied by the first coefficient. 28/2=>14 multiplies + 1 for center => 15 multiplies/5 multiplies/clock => 3 clocks/filter result. (The test bench only sends coefficient 0-14, 0 and 28 are the same value)

- Project Steps:
  - 29 samples where each sample is 48 bits (24 real and 24 imaginary)
  - 1392 FFs (29 samples * 48 bits)
  - 810 coefficients
  - 2202 samples (1392 samples + 810 coefficients)

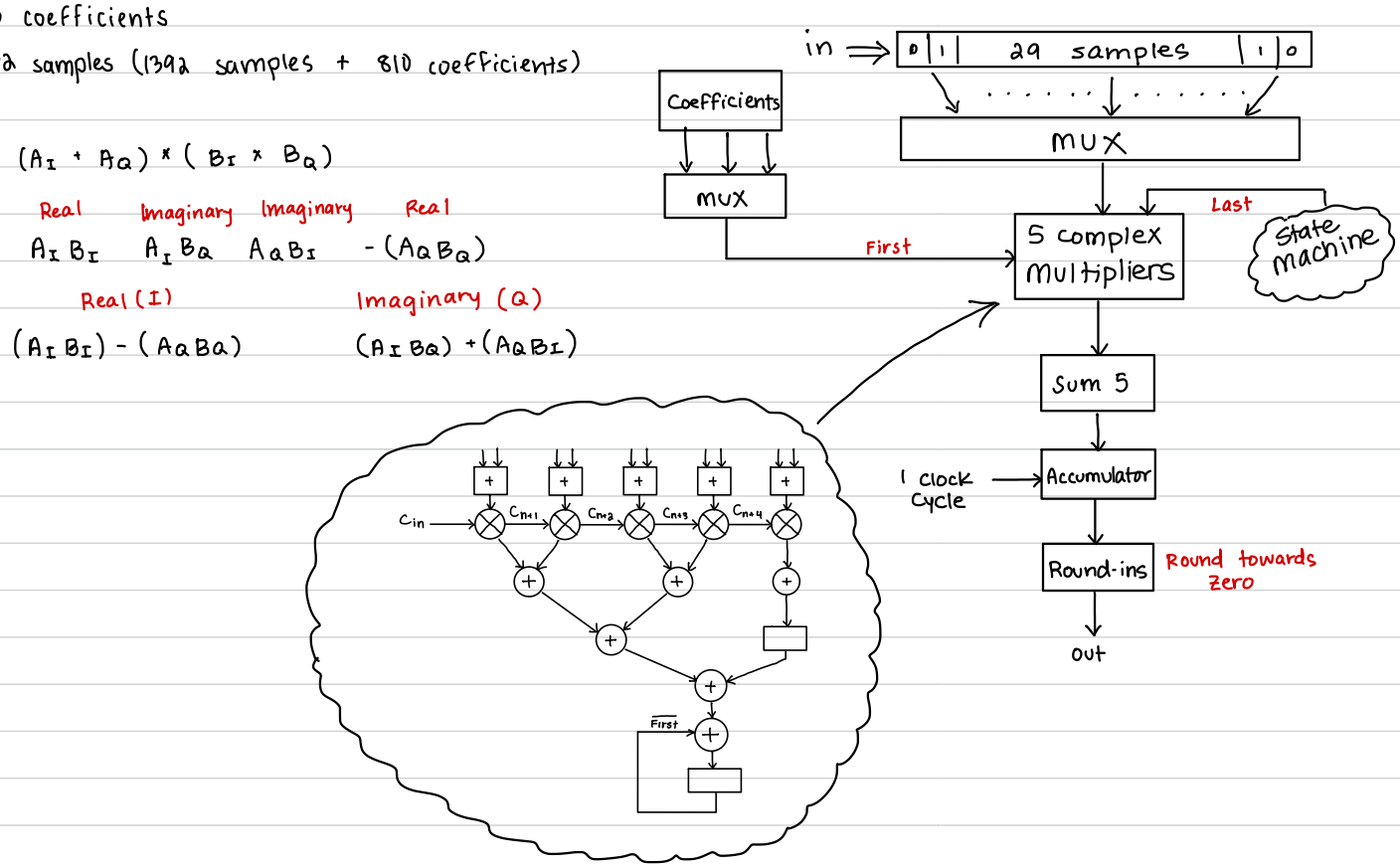
$(A_I + A_Q) * (B_I * B_Q)$

Real Imaginary Imaginary Real

$A_I B_I \quad A_I B_Q \quad A_Q B_I \quad -(A_Q B_Q)$

Real (I): $(A_I B_I) - (A_Q B_Q)$

Imaginary (Q): $(A_I B_Q) + (A_Q B_I)$



Interface:

The design will consist of a single module named fir29. This module may include other modules as desired. The DW02_mult modules will be included in the test bench.

Name	Width	Dir	Comment
Clk	1	In	Rising edge clock
Reset	1	In	Active high reset signal
PushIn	1	In	A sample push signal
StopIn	1	Out	The input cannot accept a sample
SampI	24	In	The real part of the sample in 1.23 format
SampQ	24	In	The imaginary part of the sample in 1.23 format
PushCoef	1	In	Indicates loading a coefficient
CoefAddr	5	In	The coefficient location 0-15 valid
CoefI	27	In	The coefficient Real part 3.24 format
CoefQ	27	In	The coefficient Imaginary part 3.24 format
PushOut	1	Out	Data out of the filter.
FI	32	Out	Real filtered output 8.24 format
FQ	32	Out	Imaginary filtered output 8.24 format

TEAM MEETING NOTES / TO DO:

4/12/23

- Share notes
- Create list of questions
- Create list of possible approaches
- Review together on Monday (4/17) @ 2:30 PM