EE 445S Real-Time DSP Laboratory – Prof. Brian L. Evans

Discussion on YouTube at https://www.youtube.com/watch?v=usu5Yp6EQfQ (32:30 to 47:39)

Computational Complexity of Implementing a Tapped Delay Line on the C6700 DSP

To compute one output sample $y[n]$ of a finite impulse response filter of N coefficients (h_0, h_1, ... h_{N-1}) given one input sample $x[n]$ takes N multiplication and N-1 addition operations:

$$y[n] = h_0\, x[n] + h_1\, x[n\text{-}1] + \ldots + h_{N-1}\, x[n - (N\text{-}1)]$$

Two bottlenecks arise when using single-precision floating-point (32-bit) coefficients and data on the C6700 DSP. First, only one data value and one coefficient can be read from internal memory by the CPU registers during the same instruction cycle, as there are only two 32-bit data busses. The load command has 4 cycles of delay and 1 cycle of throughput. Second, accumulation of multiplication results must be done by four different registers because the floating-point addition instruction has 3 cycles of delay and 1 cycle of throughput. Once all of the multiplications have been accumulated, the four accumulators would be added together to produce one result. The code below does not use looping, and does not contain some of the necessary setup code (e.g. to initiate modulo addressing for the circular buffer of past input data).

Cycle	Instruction
1	**LDW *x*[*n*] || LDW h_0**
2	LDW $x[n\text{-}1]$ || LDW h_1 || ZERO accumulator0
3	LDW $x[n\text{-}2]$ || LDW h_2 || ZERO accumulator1
4	LDW $x[n\text{-}3]$ || LDW h_3 || ZERO accumulator2
5	LDW $x[n\text{-}4]$ || LDW h_4 || ZERO accumulator3
6	LDW $x[n\text{-}5]$ || LDW h_5 || **MPYSP *x*[*n*], h_0, product0**
7	LDW $x[n\text{-}6]$ || LDW h_6 || MPYSP $x[n\text{-}1]$, h_1, product1
8	LDW $x[n\text{-}7]$ || LDW h_7 || MPYSP $x[n\text{-}2]$, h_2, product2
9	LDW $x[n\text{-}8]$ || LDW h_8 || MPYSP $x[n\text{-}3]$, h_3, product3
10	LDW $x[n\text{-}9]$ || LDW h_9 || MPYSP $x[n\text{-}4]$, h_4, product4 || **ADDSP product0, accumulator0, accumulator0**
11	LDW $x[n\text{-}10]$ || LDW h_{10} || MPYSP $x[n\text{-}5]$, h_5, product5 || ADDSP product1, accumulator1, accumulator1
12	LDW $x[n\text{-}11]$ || LDW h_{11} || MPYSP $x[n\text{-}6]$, h_6, product6 || ADDSP product2, accumulator2, accumulator2
13	LDW $x[n\text{-}12]$ || LDW h_{12} || MPYSP $x[n\text{-}7]$, h_7, product7 || ADDSP product3, accumulator3, accumulator3
14	LDW $x[n\text{-}13]$ || LDW h_{13} || MPYSP $x[n\text{-}8]$, h_8, product8 || ADDSP product4, accumulator0, accumulator0
15	…

The total number of execute cycles to compute a tapped delay line of N coefficients is the delay line length (N) + LDW throughput (1) + LDW delay (4) + MPYSP throughput (1) + MPYSP delay (3) + ADDSP throughput (1) + ADDSP delay (3) + adding four accumulators together (8) + STW throughput (1) + STW delay (4) = N + 26 cycles. If we were to include two instructions to set up the modulo addressing for the circular buffer, then the total number of execute cycles would be N + 28 cycles.