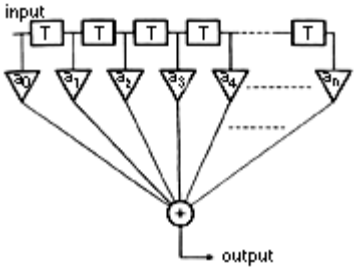
FIR FILTER HARDWARE

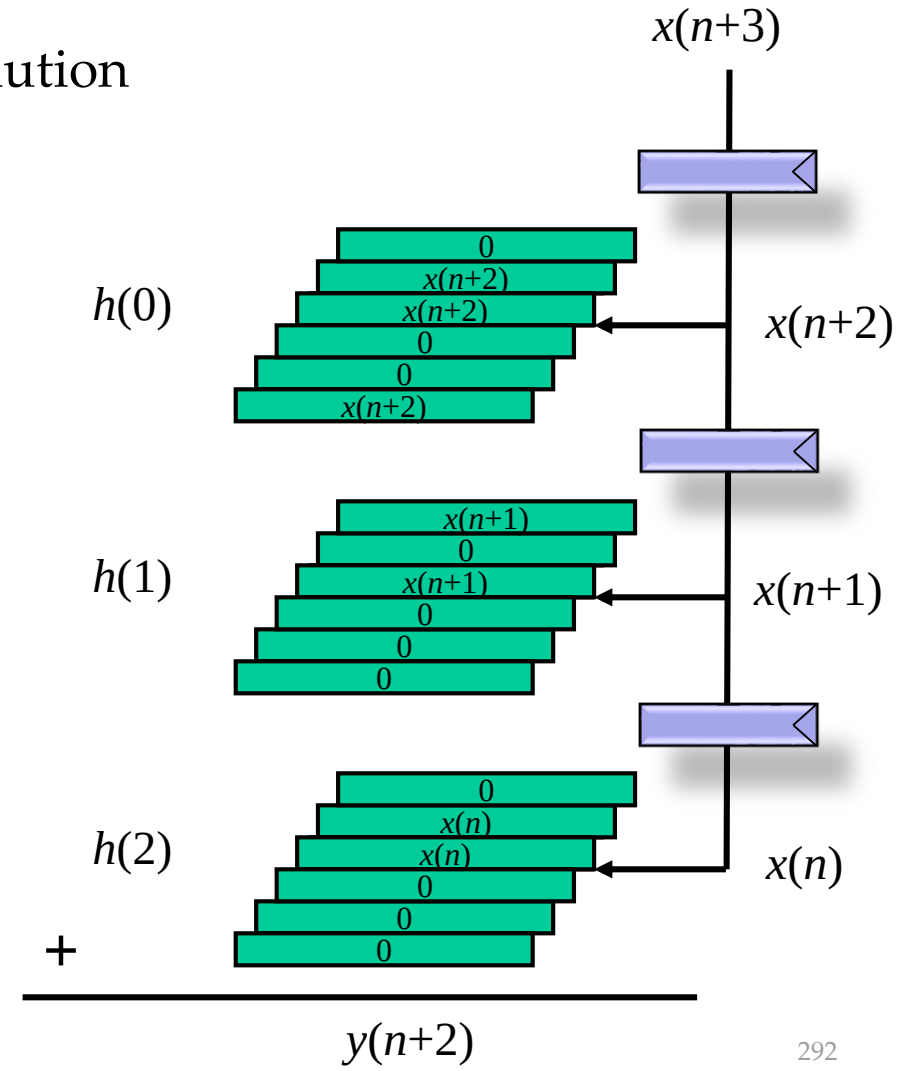
FIR Filter Review

- FIR filters are calculated by convolution

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k)$$

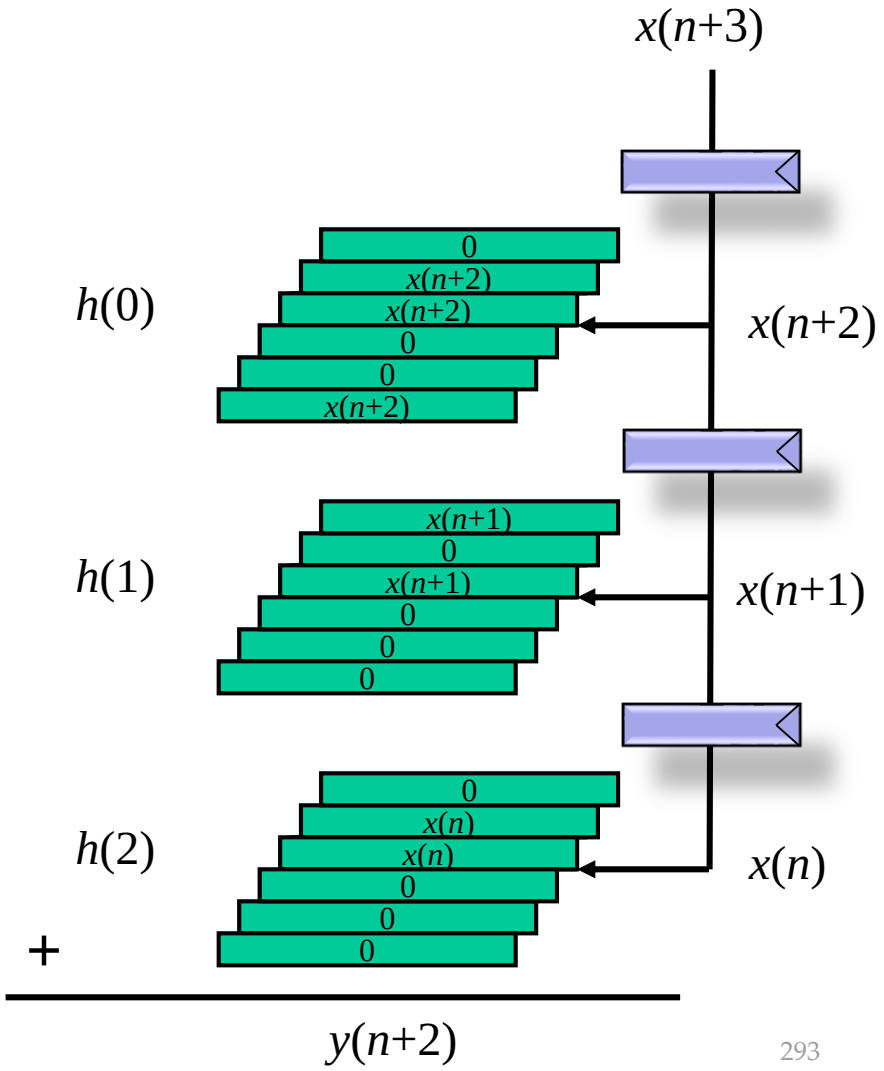
- $y(n) = x_n\, h_2 + x_{n+1}\, h_1 + x_{n+2}\, h_0$
- A "full" non-iterative direct-form FIR filter with a throughput of one sample per cycle adds the products of each coefficient times a delayed version of the input





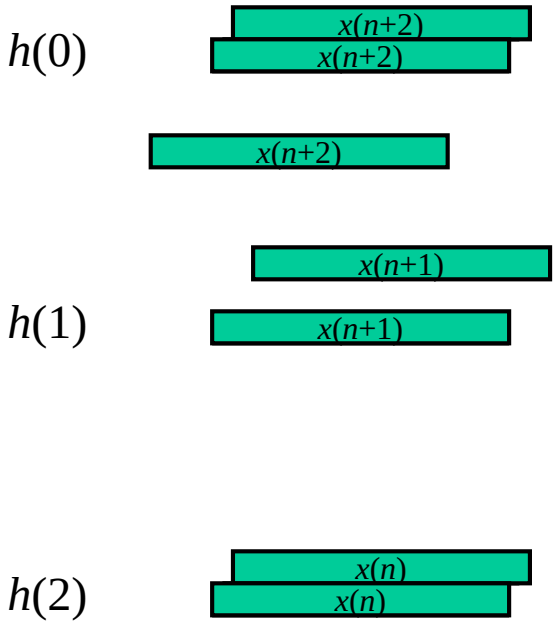
FIR Filter Example

- For direct-form FIR filters which add multiple products into an output sum, an efficient implementation adds all products in a single large multiple-input adder using the efficient carry-save adders → carry-propagate adder structure



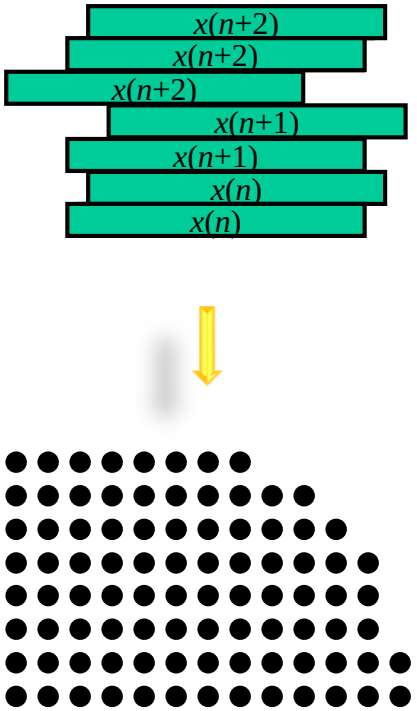
FIR Filter Example

- In the common case when the *h* coefficients are fixed, there is of course no change and no downside if the "always zero" partial products are removed



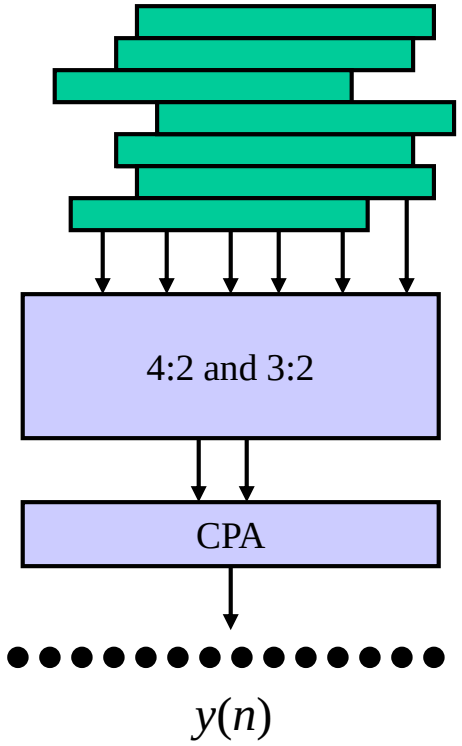
FIR Filter Example

- In the single large multiple-input adder, partial products from all filter taps get merged with each other in one large "pile of dots" where we no longer need to distinguish from which $x(n)$ and $h(n)$ a dot came
- The dots are added with a single carry-save adder structure in a manner similar to how multiplier partial-products are reduced



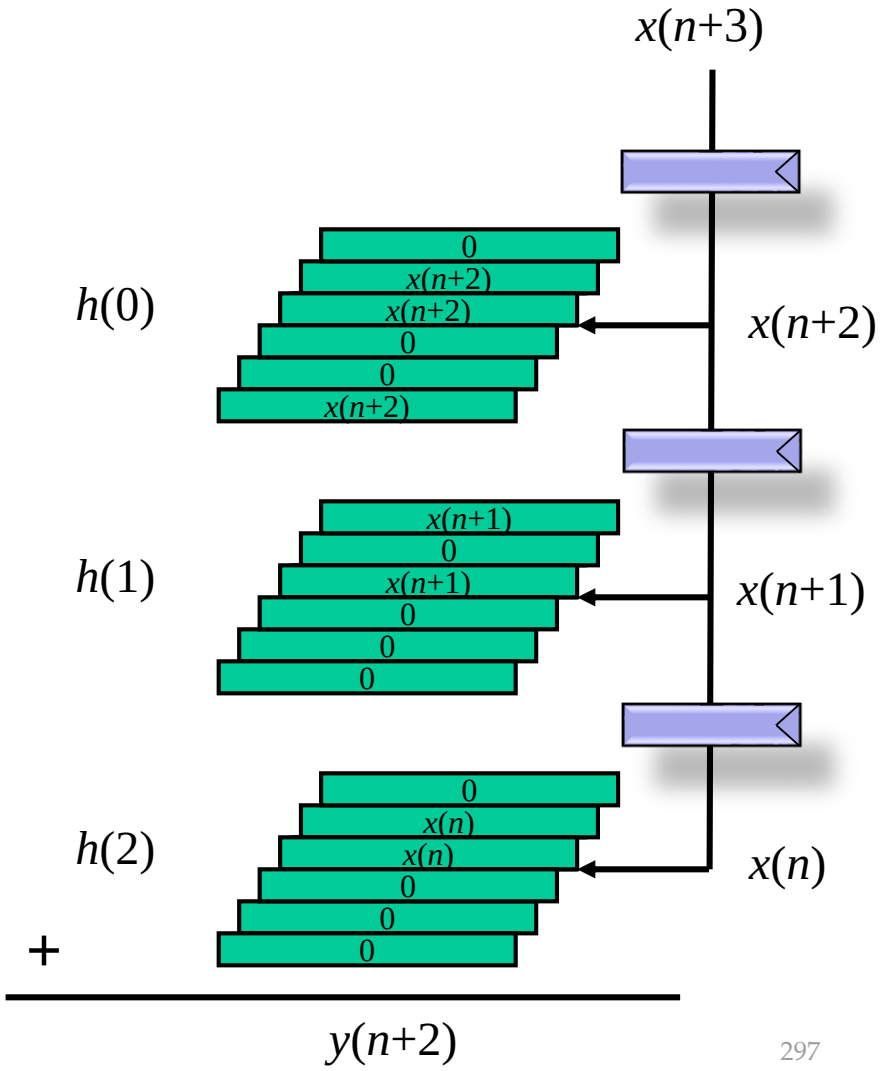
FIR Filter Example

- A carry-propagate adder finally reduces the y filter output to a single word



FIR Filter HW Reduction

- If we can scale all coefficients by the same amount
  - It may be possible to reduce a filter's complexity significantly if we can find $h(n)$ values with more zero partial products
  - The frequency response will be unchanged
  - The overall filter output magnitude *is* changed
  - Must watch out for:
    - Overflow—if the filter's output is scaled larger
    - Quantization noise—if the filter's output is scaled smaller



FIR Filter Scaling

Partial products: 2 2 4 2 2

- If `coeffs = [9 18 45 18 9]`
  note that `0.889 x coeffs = [8 16 40 16 8]`

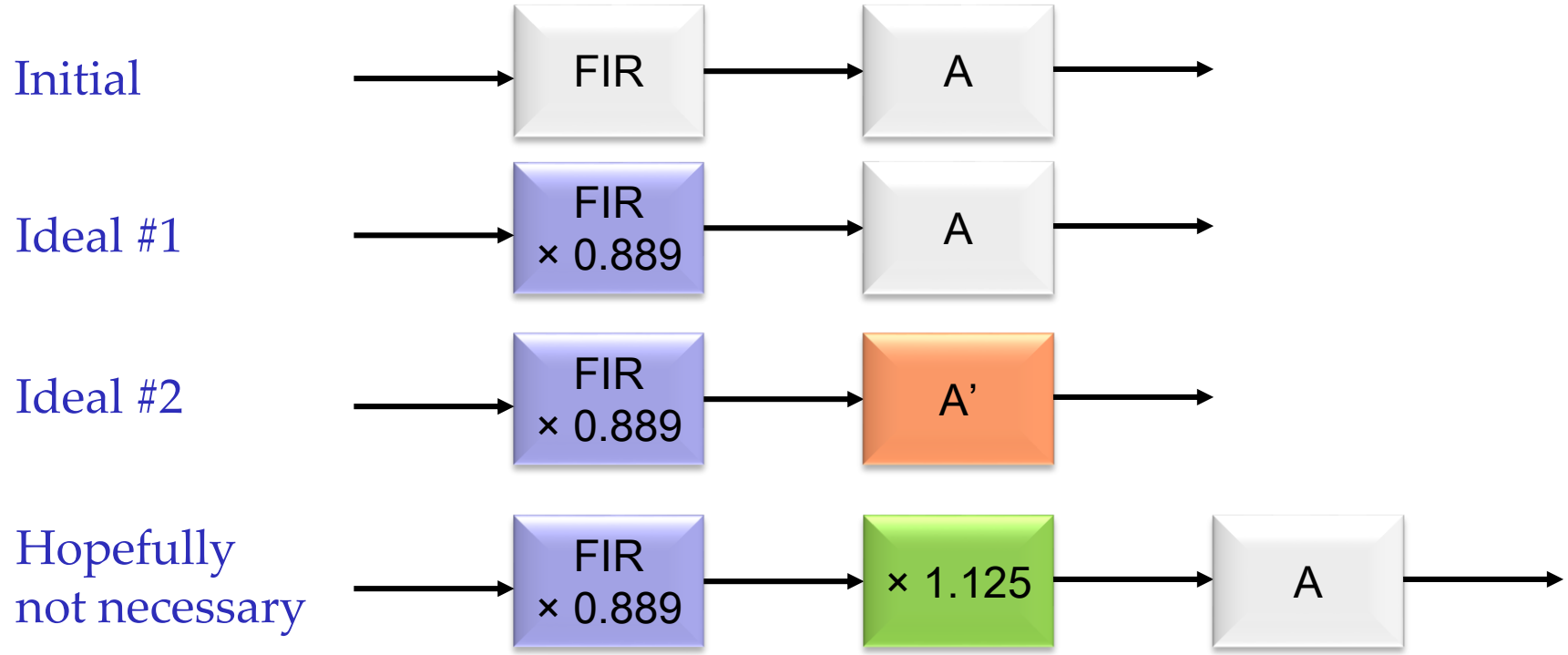
Partial products: 1 1 2 1 1

12 partial products

6 partial products

FIR Filter Scaling

- Often, scaling of a filter can be
  - ignored, or
  - accommodated or compensated elsewhere in the signal path



FIR Output Range Issues

- Worst-case inputs: maximum pos/neg samples
  - Signs of input samples match coefficient signs
  - Signs of input samples are opposite of coefficient signs
- As filter lengths increase, the peak-to-average ratio can get very large. Possible solutions include:
  - Widen the widths of all words so that the full range output can be accommodated
    - Can be unnecessary and wasteful of hardware and energy
  - Overflow
    - Almost certainly risky and a very bad idea
  - Saturate
    - Some distortion if the input is greater than the saturation level
  - Compression
    - Lower distortion when the signal enters saturation but the signal is distorted before it reaches the saturation level