2023 Spring VLSI DSP Homework Assignment #3

Due date: 2024/4/23

Q1. For the convolution DG shown in Figure 1, assume each DG node performs a multiply-and-accumulate operation, where b_i's stand for parameters, and $u(\cdot)$'s indicate input samples.

(a) Which of the following sets of scheduling and projection are permissible?

i. $\mathbf{s} = [1\ 0]^T$, $\mathbf{d} = [1\ 0]^T$

ii. $\mathbf{s} = [0\ 1]^T$, $\mathbf{d} = [1\ 0]^T$

iii. $\mathbf{s} = [1\ 1]^T$, $\mathbf{d} = [1\ 0]^T$

iv. $\mathbf{s} = [1\ -1]^T$, $\mathbf{d} = [0\ 1]^T$

(b) derive the mapping for each permissible set

(c) reverse the direction of data accumulation in the DG, derive a systolic array mapping (all inter-PE data links should have at least one delay element) for it

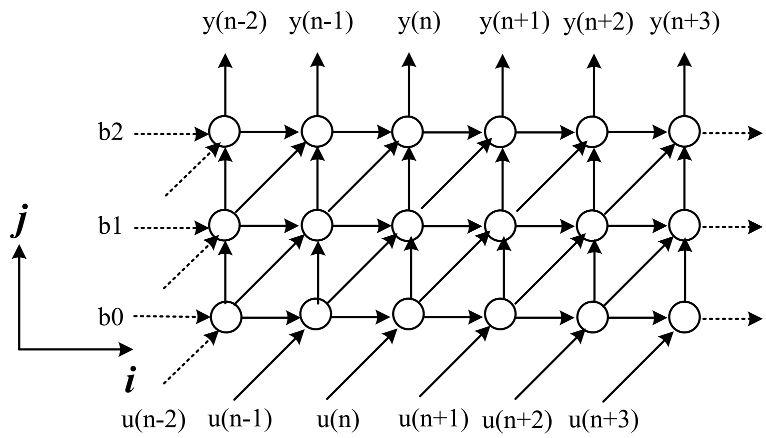

Figure 1

Q2. For the given ARMA filter

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + b_2 \cdot x(n-2) + b_3 \cdot x(n-3) + a_1 \cdot y(n-1) + a_2 \cdot y(n-2)$$

(a) Please derive its dependence graph

(b) Determine the scheduling and projection vectors and derive a mapping of direct form

(c) Determine the scheduling and projection vectors and derive a mapping of transpose form

Q3. Vector quantization design

Given an input k-dimensional column vector $\mathbf{r}_{k\times1}$, and a codebook $\mathcal{B}=\{\mathbf{b}_i \mid i = 1 \sim N\}$ consists of N k-dimensional column vectors, vector quantization (VQ) is to find a vector in that codebook that has the shortest Euclidean distance from the input vector $\mathbf{r}_{k\times1}$. The Euclidean distance between two vectors is defined as

$$d'(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}|| = \sqrt{\textstyle\sum_{j=0}^{k-1}(x_j - y_j)^2} \tag{1}$$

For simplicity, we may use the square distance instead.

$$d(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||^2 = \textstyle\sum_{j=0}^{k-1}(x_j - y_j)^2 \tag{2}$$

Let $\mathbf{x} = \mathbf{r}$ and $\mathbf{y} = \mathbf{b}_i$, Eq(2) can be rewritten as $d(\mathbf{r}, \mathbf{b}_i) = \|\mathbf{r}\|^2 - 2\mathbf{r}^t\mathbf{b}_i + \|\mathbf{b}_i\|^2$. Since $\|\mathbf{r}\|^2$ is a constant term in all distance calculations, and $\|\mathbf{b}_i\|^2$ can be precomputed, VQ calculation can be expressed as

$$\arg\{\min\{c_i - \mathbf{r}^t\mathbf{b}_i \mid i = 1, N\}\}, \tag{3}$$

where $c_i = \|\mathbf{b}_i\|^2 / 2$. In other words, VQ can be accomplished by calculating $c_i - \mathbf{r}^t\mathbf{b}_i$, for all $\mathbf{b}_i$'s in the codebook, and recording the one with the smallest distance. Note that $\mathbf{r}^t\mathbf{b}_i$ is an inner product operation, and c_i can be input from a pre-computed table.

(a) Please draw the DG of the $y(i) = c_i - \mathbf{r}^t\mathbf{b}_i$ for i = $1{\sim}N$. For simplicity, assume the vector dimension k is 4. In each iteration, c_i and $\mathbf{b}_i$ are regarded as input and $y(i)$ is the output.

(b) Select a scheduling and projection scheme to obtain its systolic array design (at least one delay element in every inter-processor data link).

(c) Add a comparator module so that the vector index i corresponding to the minimum Euclidean distance can be obtained at the end of N iterations.