Contents

Introduction

Pipelining of FIR Digital Filters

Parallel processing

Low power Design
A FIR Filter is defined as follows:

\[ y(n) = \sum_{k} a_k x(n-k) \]

For example:

\[ y(n) = a \cdot x(n) + b \cdot x(n-1) + c \cdot x(n-2) \]
Another way to represent DSP signals is by SFG. SFG is a collection of nodes and directed edges.

- Node represents a computations or tasks
  - Node is basically a multiplier or an adder.
- Edge $(j,k)$ is a branch from node $j$ to node $k$ and represents a linear transformation from the signal at node $j$ to the signal at node $k$.
  - Edge are restricted to constant gain multipliers, delay elements
Example
Exercise

- Perform the block diagram and SFG of the following examples:

\[ y(n) = x(n) - 2 \cdot x(n-1) - 2 \cdot x(n-2) + x(n-3) \]

\[ y(n) - 2 \cdot y(n-1) = x(n) - x(n-1) \]
Critical Path

- Critical path is the longest path in the SFG.

As can be seen in this example, the longest path is marked in red and is composed by

- 1 multiplier
- 2 additions
Sampling Frequency

- The critical path determines the minimum time required for processing a new sample.
- The sampling period for a digital FIR filter is given by

\[ T_s \geq T_M + 2T_A \]

- Where \( T_M \) is the time taken by a multiplier and \( T_A \) is the time taken by the adder.
Sampling frequency

- The sampling frequency is given by

\[ f_s \leq \frac{1}{T_M + 2T_A} \]

- Sampling frequency is limited by the critical path

- The direct-form structure can only be used when this equation can be satisfied.

- Some real-time applications demand faster input rate (sampling frequency)
## Pipelining and Parallel Processing

<table>
<thead>
<tr>
<th>Pipelining</th>
<th>Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipelining transformation leads to a reduction in the critical path, this</td>
<td>Multiple outputs are compute in parallel for in a clock period</td>
</tr>
<tr>
<td>reduction leads to</td>
<td></td>
</tr>
<tr>
<td>▶ Increase clock speed</td>
<td>▶ This approach can also be use for reduction of power consumption</td>
</tr>
<tr>
<td>▶ Increase sampling frequency</td>
<td></td>
</tr>
<tr>
<td>▶ Reduce power consumption</td>
<td></td>
</tr>
</tbody>
</table>
### Pipelining and Parallel Processing

<table>
<thead>
<tr>
<th>Pipelining</th>
<th>Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Reduction of the critical path is achieved by introducing pipelining latches</td>
<td>- Parallel processing reduces the sampling rate by replicating hardware so that several inputs can be processes at the same time</td>
</tr>
</tbody>
</table>
Pipelining of FIR Digital Filters

- Pipelining cuts the critical path by introducing pipelining latches.
- Example

\[
x(n) \rightarrow z^{-1} \rightarrow z^{-1} \rightarrow D \rightarrow + \rightarrow y(n)
\]

- Critical path has change from \( T_M + 2T_A \) to \( T_M + T_A \).
Pipelining of FIR Digital Filters

- The schedule of events for the pipelining system is shown in the following table:

<table>
<thead>
<tr>
<th>Clock</th>
<th>Input</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x(0)</td>
<td>a x(0) + b x(-1)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>x(1)</td>
<td>a x(1) + b x(0)</td>
<td>a x(0) + b x(-1)</td>
<td>c x(-2)</td>
<td>y(0)</td>
</tr>
<tr>
<td>2</td>
<td>x(2)</td>
<td>a x(2) + b x(1)</td>
<td>a x(1) + b x(0)</td>
<td>c x(-1)</td>
<td>y(1)</td>
</tr>
<tr>
<td>3</td>
<td>x(3)</td>
<td>a x(3) + b x(2)</td>
<td>a x(2) + b x(1)</td>
<td>c x(0)</td>
<td>y(2)</td>
</tr>
</tbody>
</table>

- At any time, 2 consecutive outputs are computed in an interleaved manner.
In an $M$-level pipeline system, the number of delay elements in any path from the input to the output is $(M-1)$ greater than the original system.

Pipelining reduces the critical path but increases the latency.

Latency is the difference in the availability of the first output data.

The drawbacks of pipelining is the increase in the number of latches and the system latency.
Pipelining of FIR Digital Filters

- The speed of an architecture is limited by the longest path between 2 latches.

- The longest path can be reduced by placing pipeline latches.

- The pipelining latches can only be placed across feed-forward cutset of the graph.
  - Feed-forward cutset is a set of edges (all on the forward direction) that if they are removed the graph becomes disjoint.
The critical path of the original 3-tap FIR filter can be reduced without introducing any pipelining latch by transposing the structure.

Transposing theorem

- Reversing the direction of all the edges in a given SFG and interchanging the input and output ports preserves the functionality of the system.
Transposing

\[ x(n) \rightarrow z^{-1} \rightarrow b \rightarrow z^{-1} \rightarrow y(n) \]

\[ y(n) \rightarrow z^{-1} \rightarrow b \rightarrow z^{-1} \rightarrow x(n) \]
Transposing leads to a data-broadcast structure where data are not stored but are broadcast to all multipliers simultaneously.

Critical path is $T_M + T_A$
Let $T_M = 10$ units and $T_A = 2$ units and the desired clock period is $(T_M + T_A)/2 = 6$ units.

Then, the TM must be split into 2 smaller units with processing time 6 units and 4 units.
Parallel Processing

- Pipeline and parallel techniques are dual of each other.

- Pipeline processes data in interleaved fashion while parallel systems process more than one data using duplicated hardware.

- The system must be transformed to receive multiple data to be process by the parallel architecture.
Consider the Single-Input-Single-Output (SISO) filter describe as follows:

\[ y(n) = a \cdot x(n) + b \cdot x(n-1) + c \cdot x(n-2) \]

To obtain a parallel processing structure, the SISO system must be converted into a Multiple-Input-Multiple-Output (MIMO) system.
The 3-tap FIR filter can be transformed with a level of parallelism $L = 3$ as follows:

\[
\begin{align*}
y(3k) &= a \cdot x(3k) + b \cdot x(3k - 1) + c \cdot x(3k - 2) \\
y(3k + 1) &= a \cdot x(3k + 1) + b \cdot x(3k) + c \cdot x(3k - 1) \\
y(3k + 2) &= a \cdot x(3k + 2) + b \cdot x(3k + 1) + c \cdot x(3k)
\end{align*}
\]

where $k$ is the clock cycle.

At $k$-clock cycle 3 samples are processed.
MIMO Structure

\[
\begin{align*}
x(3k) & 
\downarrow \quad a \\
\quad \quad \quad & \quad z^{-1} \\
\quad \quad \quad & + \\
\quad \quad \quad & \quad z^{-1} \\
\quad \quad \quad & + \\
\quad \quad \quad & \quad b \\
\quad \quad \quad & \quad c \\
\quad \quad \quad & \downarrow \\
\quad \quad \quad & y(3k) \\
\end{align*}
\]

\[
\begin{align*}
x(3k+1) & 
\downarrow \quad a \\
\quad \quad \quad & \quad z^{-1} \\
\quad \quad \quad & + \\
\quad \quad \quad & \quad z^{-1} \\
\quad \quad \quad & + \\
\quad \quad \quad & \quad b \\
\quad \quad \quad & \quad c \\
\quad \quad \quad & \downarrow \\
\quad \quad \quad & y(3k+1) \\
\end{align*}
\]

\[
\begin{align*}
x(3k+2) & 
\downarrow \quad a \\
\quad \quad \quad & \quad z^{-1} \\
\quad \quad \quad & + \\
\quad \quad \quad & \quad z^{-1} \\
\quad \quad \quad & + \\
\quad \quad \quad & \quad b \\
\quad \quad \quad & \quad c \\
\quad \quad \quad & \downarrow \\
\quad \quad \quad & y(3k+2) \\
\end{align*}
\]
The critical path of the block or parallel processing system has remained unchanged and the clock period $T_{clk}$ must satisfy

$$T_{clk} \geq T_M + 2T_A$$

But since 3 samples are processed in 1 clock cycle, the sampling frequency is given by

$$T_s = \frac{1}{3} T_{clk} \geq \frac{1}{3} (T_M + 2T_A)$$
Parallel FIR Filter

Sample Period $T_{CLK}/4$

Clock Period $T_{CLK}$

Serial-to-Parallel Converter

MIMO System

Parallel-to-Serial Converter

$y(n)$

$x(n)$
Serial to Parallel Converter

\[ x(n) \xrightarrow{T/4} D \xrightarrow{T/4} D \xrightarrow{T/4} D \]

\[ x(4k+3) \xrightarrow{T} x(4k+2) \xrightarrow{T} x(4k+1) \xrightarrow{T} x(4k) \]
Parallel to Serial Converter
Parallel and Pipelining Processing

- Fine-grain pipelining can be applied to reduce the critical path.

- Combining parallel structure with fine-grain pipelining applied to the multiplier, the sample period has been further reduced to

\[ T_s = \frac{1}{LM} T_{clk} \geq \frac{1}{3 \cdot 2} (T_M + 2T_A) \]
Why apply parallelism?

- There is a fundamental limit to pipelining imposed by the input/output bottleneck.

- If the input-pad delay, output-pad delay and the wire delay between two chips is 8 nsec, then $T_{\text{CLK}}$ has to be greater or equal to 8 nsec.
If the communication time is greater than the processing time of the critical path, then the system is communication bounded.

Then, pipelining can be used only to the extent such that the critical path computation is limited by the communication or I/O bound, and once this has been reached, pipelining can no longer increase the speed.
Low Power

Pipeling and parallel processing for low power
Low power

- The two main advantages of using pipeling and parallel processing are:
  - Higher speed
  - Lower power

- The propagation delay is associated with charging and discharging of various gate and stray capacitances in the critical path

\[ T_{pd} = \frac{C_c V_s}{k(V_s - V_t)^2} \]
The power consumption of a CMOS circuit can be estimated using the following equation:

\[ P = C_{\text{total}} V_s^2 f \]

- \( C_{\text{total}} \) denotes the total capacitance of the circuit
- \( V_s \) supply voltage
- \( f \) is the clock frequency of the circuit
Pipelining for low power

- Let's define the power consumed by a FIR filter without pipelining as

\[ P_{seq} = C_{total} V_s^2 f \]

- Consider an M-level pipeline system, where the critical path is reduced to \(1/M\), then \(C_c\) is reduced to \(C_c/M\) for a single clock cycle.

- In the same time that \(C_c\) was charge/discharge, now only a fraction of it should be charge/discharge.
Pipeline for low power

- Then, the supply voltage can be reduced by $\beta$, where $0 < \beta < 1$
- The power consumption of the pipeline filter will be

$$P_{pip} = C_{total} \beta^2 V_s^2 f = \beta^2 P_{seq}$$

- The value of $\beta$ can be determined by examining the propagation delay
Propagation delay

- The propagation delay of the original filter is

\[ T_{seq} = \frac{C_c V_s}{k(V_s - V_t)^2} \]

- While the propagation delay of the pipeline filter is

\[ T_{pip} = \frac{C_c \beta V_s}{M} \frac{\beta V_s}{k(\beta V_s - V_t)^2} \]
The same clock speed is maintained for both filters, therefore the following equation is maintained

\[ M(\beta V_s - V_t)^2 = \beta (V_0 - V_t)^2 \]

Then \( \beta \) is obtained, the reduction of power consumption can be computed using

\[ P_{pip} = C_{total} \beta^2 V_s^2 f = \beta^2 P_{seq} \]
Suppose that the capacitance of the multiplier is 5 times the capacitance of the adder.

The M1 has 3 times and M2 has 2 times the adder capacitance.

$V_t = 0.6\,\text{V}$ and $V_s = 5.0\,\text{V}$

What is the supply voltage of the pipeline filter?

What is the power consumption of the pipeline filter?
In parallel processing the total capacitance is not reduced, rather is increase by \( L \) times.

In order to maintain the same data rate, the clock period must be increased to \( LT_{\text{seq}} \).

Then, there is more time to charge the same capacitance.

Therefore, the supply voltage can be reduced to \( \beta V_s \).
Propogation delay

The propagation delay of the original filter is

\[ T_{seq} = \frac{C_c V_s}{k(V_s - V_t)^2} \]

While the propagation delay of the parallel filter is

\[ T_{par} = L T_{seq} = \frac{L C_c \beta V_s}{k(\beta V_s - V_t)^2} \]
The same clock speed is maintained for both filters, therefore the following equation is maintained

\[ L(\beta V_s - V_t)^2 = \beta (V_0 - V_t)^2 \]

Then \( \beta \) is obtained, the reduction of power consumption can be computed using

\[ P_{par} = LC_c (\beta V_s)^2 \frac{f}{L} = \beta^2 C_{charge} V_s^2 f = \beta^2 P_{seq} \]
Example

- Consider a 4-tap FIR filter and a L=2 parallel filter.
- Assume that the multiplication operation takes 8 ut and the addition 1 ut.
- Assume that the capacitance of the multiplier is 8 times that of an adder.
- The two architecture are operated at a sample period of 9 ut.
- \( V_t = 0.45 \text{V} \) and \( V_s = 3.3 \text{V} \)
  - What is the supply voltage of the 2-parallel filter?
  - What is the power consumption of the 2 parallel filter?
Combining Parallel and Pipelining

- Pipeling reduces the capacitance to be charged/discharge in 1 clock period.

- Parallel processing increases the clock period for charging/discharging the original capacitance.
Propagation delay

- The propagation delay of the original filter is

\[ T_{seq} = \frac{C_c V_s}{k(V_s - V_t)^2} \]

- While the propagation delay of the parallel filter is

\[ T_{pp} = L T_{pip} = \frac{L(C_c/M0)\beta V_s}{k(\beta V_s - V_t)^2} \]
The same clock speed is maintained for both filters, therefore the following equation is maintained:

\[ ML(\beta V_s - V_t)^2 = \beta(V_0 - V_t)^2 \]

Then \( \beta \) is obtained, the reduction of power consumption can be computed using:

\[ P_{pp} = \beta^2 P_{seq} \]
Example

- Considering \( L = M = 2 \), \( V_s = 5V \) and \( V_t = 0.6V \). Then \( \beta = 0.4 \)

- The power consumption is reduced by a factor of 0.16; \( P_{pp} = 0.16P_{seq} \)

- However, there is a limit imposed by the threshold voltage. The supply voltage cannot be lower than \( V_t \)