CHAPTER 3

VARIABLE PRECISION TWO-DIMENSIONAL
FINE GRAIN PIPELINE

The variable precision two-dimensional fine-grain pipeline technique is applied for the pipelined digital system. This technique reduces the power consumption of the system based on the nature of input signal. This chapter discusses the design of multiplier and FIR filter using this technique.

3.1 INTRODUCTION

The digital multipliers are basic building blocks of digital signal processors. Digital multipliers are the most critical arithmetic functional unit in many DSP applications, viz. Fourier Transform (FT), Discrete Cosine Transform (DCT), filtering, etc. Array and parallel multipliers are important due to their high execution speed and throughput. The energy consumption of such circuits designed using CMOS logic is proportional to the number of transitions. Hence, unwanted transitions in the circuit must be reduced to reduce the power consumption.

The proposed variable precision two-dimensional fine-grain pipeline technique is applied on the pipeline to obtain better power reduction by reducing the switching activities. This technique is to gate the clock to the registers in both vertical direction and horizontal direction to the data flow. In designing the pipelined multiplier, the effectiveness of this multiplier lies in its capacity to increase with the growth of the multiplication length. The
gating signals are generated from the current precision information of the input data. These gating signals are combined with system clock to generate sub-clocks, which are connected to the corresponding registers in all pipeline stages. This is suitable for both signed and unsigned multipliers. Then FIR filter is designed and implemented using Xilinx Spartan 3E FPGA. The power consumption is analyzed for the circuit designed using Xilinx XPower Analyzer tool.

### 3.2 TWO-DIMENSIONAL PIPELINE GATING

In two-dimensional pipeline gating technique discussed by Di et al (2003), clock gating used in data path direction in the pipeline. The system clock is distributed to the different sections as a sub clock based on the precision of the input data. Each sub-clock is connected to one pipeline stage and it drives all registers in that stage. Based on the input data, some of the register stages not used in the calculation of result are disabled and results are diverted to the output through multiplexer as shown in the Figure 3.1. This reduces the pipeline latency. As there are no transitions in the masked stage the power consumption also reduced. In general, the above method provides power awareness in the digital system.

![Figure 3.1 Two-Dimensional Pipeline Gating Technique](image-url)
This method gives the power awareness to non-pipelined digital system. However, for implementing the pipeline concept, latency must be known and to be fixed. The proposed technique fixed the latency for reduced power consumption in pipelined digital systems. There are two extra features are added in the proposed technique, namely, variable precision intra pipeline-stage gating and pipelining of sub clocks and bypass signals to maintain the latency of the system.

3.3 VARIABLE PRECISION TWO-DIMENSIONAL FINE GRAIN PIPELINE FOR MULTIPLIER

This study proposes to use variable precision two-dimensional fine grain pipeline scheme to achieve more power saving. It can greatly improve the power awareness in pipelined digital systems.

The proposed scheme is shown in Figure 3.2. Under certain condition, pipeline stage-4 can be disabled and some of the registers in previous stages can also be disabled if the data flow through the stage-4 is not required to process. These registers can be disabled by using the sub clock C_3. (The register is represented as ‘R’ in Figure 3.2).

For the same reason, if stage-3 needs to be disabled, some registers in stage-1 and stage-2 can also be disabled. The total numbers of transitions are further reduced. As the number of registers in each stage as well as the total number of stages in the pipeline (pipeline depth) increase, this further benefit becomes more and more significant. Sub clocks and bypass signals from different stages are pipelined and driven by the system clock. This is done in order to maintain the fixed latency and to directly used in any fine grain pipelined system.
Figure 3.2 Variable Precision Two-dimensional Pipeline Gating Technique

To design a multiplier using this technique, firstly the multiplication process is examined. The $4 \times 4$ multiplication process is displayed in Figure 3.3. Here $X$ and $Y$ are inputs while $S$ is the output. When the input precision is 4, for example, calculating $1001 \times 1111$, $S$ is generated based on all inner partial products.

If the input precision is 3, for example, calculating $0110 \times 0101$, the partial products containing $X_3$ or $Y_3$ are all zero and $S$ will have six bits instead of eight. From a reset-to-zero state, the registers need not to propagate these zeros, because the reset state of register is zero. So $C_3$ connected to these registers are used to disable these registers.

Under a certain input precision, one or more sub-clocks ($C_0$, $C_1$, $C_2$, $C_3$) may be used. The registers connected to these sub-clocks will be disabled during the calculation of results. The multiplexers will select correct outputs from the corresponding stages. For example, while performing $0001 \times 0001$, only $S_0$ has useful value. This value is selected from the stage right after the AND matrix. Except this register and the two registers in the first stage for $X_0$ and $Y_0$, all other registers do not function, because their clocks are disabled. Hence, the power dissipation is reduced significantly. The output $S_0$ is from
the first stage after the AND matrix instead of the eighth stage, thus the pipeline latency has been reduced by a factor of eight in a non-pipelined structure. To use this multiplier for fine-grain pipelined FIR filter, the sub clocks and bypass signals to the multiplexer are passes through Flip-Flops (FF) driven by system clock as shown in Figure 3.4

<table>
<thead>
<tr>
<th></th>
<th>X₃ X₂ X₁ X₀</th>
<th>x</th>
<th>Y₃ Y₂ Y₁ Y₀</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X₃Y₀</td>
<td>X₂Y₀</td>
<td>X₁Y₀</td>
</tr>
<tr>
<td></td>
<td>X₃Y₁</td>
<td>X₂Y₁</td>
<td>X₁Y₁</td>
</tr>
<tr>
<td></td>
<td>X₃Y₂</td>
<td>X₂Y₂</td>
<td>X₁Y₂</td>
</tr>
<tr>
<td></td>
<td>X₃Y₃</td>
<td>X₂Y₃</td>
<td>X₁Y₃</td>
</tr>
</tbody>
</table>

**Figure 3.3 4x4 Multiplication Process**

For example, in digital camera, when the resolution of the image is reduced, this information is sent to the processing system. Then it generates and sets up the clock gating signals based on the resolution required. Figure 3.3 shows the different masking clocks required for different partial products.

The additional area cost to implement this is very low, just a few AND gates and some multiplexers are needed. The clock gating signals are also used as the control signals for these multiplexers. HA and FA in Figure 3.4 represent half adder and full adder.

Based on the above discussion, a set of 8-bit variable precision pipeline multiplier is designed (for simplicity Figure 3.4 shows the 4 x 4 multiplier). Both pipeline gating and variable precision pipeline gating
techniques are applied to each multiplier. This multiplier is synthesized by Xilinx ISE® tool and power is analyzed using Xilinx Xpower analyzer. During the analysis, the multipliers are fed with data of different input precision. The power dissipation is recorded and compared.

![Diagram of a 4x4 Variable Precision Two-Dimensional Pipelined Multiplier](image)

**Figure 3.4 4x4 Variable Precision Two-Dimensional Pipelined Multiplier**

**Sign extension detection for 2’s complement multiplier:** In many multimedia and signal processing applications, the bit precision used for data representation is varied, depending on the amount of accuracy sought. The number of bits used to represent data directly translates to the number of quantization levels that the dynamic range can be divided into and determines the quantization error. The higher the bit precision, the greater is the accuracy.
However, when algorithms are implemented on standard DSP or general-purpose processors, computational precision is fixed. When it is implemented in Application Specific Integrated Circuits (ASIC), this is definitely significant in terms of power.

Consider the two cases exhibited in Figure 3.5. The four registers are represented with two’s complement 16-bit data (Sinha et al 2002). The sign extension bits are shaded in every register.

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 1</td>
<td>0 1 0 1 1 0 0 1 0 0 1 1 1 1 1 1</td>
</tr>
<tr>
<td>0 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0</td>
<td>0 1 1 1 1 0 1 0 1 0 0 0 1 1 1 0</td>
</tr>
<tr>
<td>1 1 0 0 1 1 0 1 0 1 0 0 0 1 1 1</td>
<td>1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1</td>
</tr>
</tbody>
</table>

**Figure 3.5 Representation of four 16-bit Registers**

It may be noticed that in case of small numbers, the Most Significant Bits (MSB) are 0 for positive quantities and 1 for negative quantities as shown in Figure 3.6.

<table>
<thead>
<tr>
<th>Rejected Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1</td>
</tr>
</tbody>
</table>

**Figure 3.6 Sign Extension Bits**
No accuracy is lost if the sign-extension bits are rejected. In the next step, the number of sign-extension bits are determined using the circuit as shown in Figure 3.7.

![Figure 3.7 Circuit Diagram for Determining Sign-Extension Bits](image)

The ‘1’ outputs of the mask determine the sign-extension bits in each register as arrayed in Figure 3.8. The number of sign-extension bits that can be rejected is equal to the minimum of the sign-extension bits among all the registers. This can be obtained simply by ANDing all the individual mask outputs from each register.

![Figure 3.8 Mask Outputs for Sign-Extension Bits](image)
3.4 APPLICATION OF VARIABLE PRECISION TWO DIMENSIONAL FINE GRAIN PIPELINE TECHNIQUE FOR FIR FILTER

Pipelining can be used to reduce the power consumption of a FIR filter as discussed in chapter 1. In pipelining, high speed and low power consumption are the two main advantages. Now consider the use of these techniques for lowering the power consumption where the sample speed need not be increased. The propagation delay \( T_{pd} \) is associated with charging and discharging of the various gate and stray capacitances in the critical path. For the CMOS circuit, the propagation delay is given in equation 3.1

\[
T_{pd} = \frac{C_{\text{charge}} V_{dd}}{k (V_{dd} - V_t)^2}
\]  

(3.1)

where, \( C_{\text{charge}} \) is charge/discharge capacitance, \( V_{dd} \) is the supply voltage and \( V_t \) is threshold voltage. The parameter \( k \) is a function of technology parameters \( \mu \), \( W/L \) and \( C_{ox} \). Where \( \mu \) is the carrier mobility, \( C_{ox} \) is the gate capacitance and \( W/L \) is width to length ratio of gate of a MOS transistor.

The power consumption of a CMOS circuit is estimated using simple first order approximations in the following equation 3.2.

\[
P = C_L V_{dd}^2 f_{\text{clock}}
\]  

(3.2)

where, \( C_L \) is Load Capacitance, \( V_{dd} \) is the supply Voltage and \( f_{\text{clock}} \) is the clock frequency.

Let, \( P_{\text{seq}} = C_{\text{total}} V_{dd}^2 f \)

(3.3)

where \( P_{\text{seq}} \) represents the power consumption of the non-pipelined filter.
It should be noted that
\[ f = \frac{1}{T_{seq}} \]  
(3.4)

where, \( T_{seq} \) is the clock period of the original non-pipelined filter.

Now consider an M-level pipelined system, where the critical path is reduced to \( 1/M \) of its original length. The capacitance to be charged/discharged in a single clock cycle is reduced to \( C_{\text{charge}}/M \).

It is noticed that the total capacitance does not change. If the same clock speed is maintained, i.e. the clock frequency \( f \) is maintained, only a fraction of the original capacitance, \( C_{\text{charge}}/M \), is being charged and discharged in the same amount of time as previously needed to charge or discharge the capacitance \( C_{\text{charge}} \) as shown in Figure. 3.9.

\[ \beta V_{dd} \]

\[ V_{dd} \]

\[ T_{seq} \]

Sequential

3 level

\[ \beta V_{dd} \]

\[ T_{seq} \]

\[ T_{seq} \]

\[ T_{seq} \]

Figure 3.9 Critical Path Lengths for Original and 3-level Pipelined Systems

It implies that the supply voltage can be reduced to \( \beta V_{dd} \), where \( \beta \) is a positive constant less than 1. Hence, the power consumption of pipelined filter will be
\[ P_{\text{pipe}} = C_L \cdot \frac{1}{2} \cdot V_{dd}^2 \cdot f = \beta^2 \cdot P_{\text{seq}} \]
(3.5)
The power consumption of the pipelined system is reduced by a factor of $\beta^2$ as compared with the original system. The propagation delay of the original filter is given by:

$$T_{seq} = \frac{(C_{\text{charge}} \cdot V_{dd})}{(k \cdot (V_{dd} - V_t)^2)}$$  \hspace{1cm} (3.6)

The propagation delay of the Pipelined filter is given by:

$$T_{pipe} = \frac{(C_{\text{charge}}/M \cdot \beta \cdot V_{dd})}{(k \cdot (\beta \cdot V_{dd} - V_t)^2)}$$  \hspace{1cm} (3.7)

It is noted that the clock period is usually set equal to the maximum propagation delay $T_{pd}$ in a circuit. Once $\beta$ is obtained, the reduced power consumption of the pipelined filter can be computed using Equation 3.5.

To improve the throughput of the FIR filter, a commonly used method is to pipeline the multipliers and adders. Since the multiplication time $T_M$ is usually much larger than the addition time $T_A$, much shorter critical path length can be achieved by carefully balancing the pipeline stages. Figure 3.10 shows the pipelining scheme for FIR filter. Each multiplier is divided into two pipeline stages. Series of registers is added between the two sub-multipliers. The time taken by each stage of the multiplier is denoted by $T_{M1}$ and $T_{M2}$, respectively and the delay time in the added registers is denoted by $T_{DR}$.

In this kind of pipelining only limited improvement in throughput is obtained. If the number of pipeline stages of the multipliers approaches infinity, the slowest stage is the adder and a very small part of multiplier. So, the length of the critical path is approaches adder delay $T_A$, which is small.
When the word length of input data and coefficients are short, $T_A$ is small enough for the FIR to operate in high sampling frequency. In recent years, the word length of FIR filter has been growing from 8 bits to 16 bits, 32 bits and now 64 bits. Under long word length condition, addition also takes significant time. In addition, pipelined multipliers, adders also become bottleneck in these FIR filters under such conditions.

Further, improve the throughput of FIR filters, the critical path in addition process needs to be shortened too. The adders, as well as multipliers need to be pipelined. Pipelining one adder changes the timing relationship between the two inputs of the next adder. Unlike pipelining multipliers, which do not change the relative timing sequence between adder inputs. For pipelining the adders, the additional delay elements need to be added in the subsequent adder and its corresponding multiplier.

A set of 8-tap FIR filter is designed. The multipliers in the filters are replaced with the proposed multiplier. The average power dissipation is significantly reduced by applying variable precision two-dimensional fine grain pipeline technique. The power reduction rate of variable precision two dimensional fine grain pipeline designs is better than that of the existing two-dimensional pipeline gating design. By maintaining the constant pipeline
latency, the proposed technique considerably reduces the power consumption. Based on the discussion given above, a set of 8-tap FIR Filter using array, pipelined gating and variable precision two-dimensional fine grain pipelines are designed. These designed circuits are implemented using Xilinx Spartan 3E FPGA and power is analyzed using Xilinx XPower analyzer.

3.5 XPOWER POWER ANALYSER

XPower is the power-analysis software used to analyse the power of the proposed system. It is used to estimate the power required for programmable logic design. It enables to interactively and automatically analyze power consumption for Xilinx FPGA and Complex Programmable Logic Devices (CPLD). Earlier in the design flow than ever, the total device power, power per-net, routed, partially routed or unrouted designs can be analyzed, all driven from a comprehensive graphic interface or command-line driven batch-mode. XPower also reads .vcd simulation file from the ModelSim family of Hardware Description Language (HDL) simulators to set estimation stimulus, reducing setup time, as well as from the additional simulators listed in simulator support. Xpower tool flow is shown in Figure 3.11.

![Figure 3.11 Xpower Design Flow](image-url)
In this piece of power analysis software XPower recognizes the following file types:

- **Design files**: Design file is an .ncd (FPGA device) or .cxt (CPLD device) file that contains information about the design.

- **Physical constraints file**: A Physical Constraints File (PCF) relates to a text file containing two separate sections: a section for those physical constraints created by the mapper and a section for physical constraints entered by the user. Temperature, voltage, max delay and time graphs are read from the physical constraints file (PCF). Therefore, for accurate power estimations, it is necessary to choose the right kind of PCF.

- **Settings file**: Specifies a settings file (*.xpwr.xml) to be used by XPower. A settings file that represents the current state of the power data, constrained by the reporting options that have been already specified. This file is generated by XPower when settings are saved and is used to restore the settings.

- **Simulation file**: Specified a simulation file (*.vcd) to be used by XPower. This file is the output of a simulation run on the design. IEEE standard .vcd files are accepted for input of simulation data.

### 3.6 RESULTS AND DISCUSSION

The 8-bit Array multiplier (Array), two-dimensional Pipeline Gating Multiplier (PGM) and the proposed Variable Precision two-dimensional Pipeline gating Multiplier (VPGM) are designed. Further these multipliers are used to design an 8-tap FIR filter and implemented using Xilinx Spartan 3E FPGA. The functionality of the filter is verified and
Figure 3.12 displays the power consumption of different multipliers for different input data precision. The illustration shows that the power consumption of the proposed multiplier is less and the power saving is 27% than the existing pipelined gating technique, when data precision approaches to 1 bit.

![Graph showing power consumption of multipliers](image)

**Figure 3.12  Power Consumption of 8-bit Multipliers**

The Figure 3.13(a) and 3.13(b) clearly shows that the proposed multiplier reduced the power consumption without changing delay and gate count as in the existing one.

![Bar chart showing delay of multipliers](image)

**Figure 3.13(a) Delay of 8-bit Multipliers**
Figure 3.13(b) Area of 8-bit Multipliers

Figure 3.14 displays the simulation window of the proposed Variable Precision two-dimensional Pipeline gating Multiplier.

Figure 3.14  Simulation Window of Variable Precision Two-Dimensional Pipeline Gating Multiplier
From the simulation result, the functionality of the multiplier is verified and it gives the expected output.

The Figure 3.15(a), Figure 3.15(b) and Figure 3.16 represents delay, area and the power consumption for 8-tap FIR Filters using Array multiplier (FIR array), FIR Filter using Two Dimensional Pipeline Gating Multiplier (FIR PGM) and proposed FIR filter using Variable Precision two-dimensional Pipeline Gating Multiplier (FIR-VPGM). All the filters are tested with the same set of input sample sequences containing the data with different sets of precision.

Figure 3.15(a) Delay in 8-tap FIR Filters

Figure 3.15(b) Implementation Area of 8-tap FIR Filters
Figure 3.16  Power consumption of an 8-tap FIR filter

From the above results and Table 3.1, it is clear that, the power consumption of the proposed FIR filter is reduced by 18 % with an additional area of 3% compared to the existing filter without causing any degradation of speed.

Table 3.1 Implementation Details for an 8-tap FIR filter

<table>
<thead>
<tr>
<th></th>
<th>FIR Filter using Array multiplier (FIR array)</th>
<th>FIR Filter using Two Dimensional Pipeline Gating Multiplier (FIR PGM)</th>
<th>FIR filter using Variable precision two dimensional pipeline gating Multiplier (FIR-VPGM).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay (ns)</td>
<td>31.72</td>
<td>12.54</td>
<td>12.25</td>
</tr>
<tr>
<td>Freq (MHz)</td>
<td>31.5</td>
<td>79.74</td>
<td>81.6</td>
</tr>
<tr>
<td>Gate Count</td>
<td>4823</td>
<td>11686</td>
<td>12036</td>
</tr>
</tbody>
</table>

The proposed FIR filter is implemented in Xilinx Spartan 3E FPGA and the implemented chip is shown in Figure 3.17.
Figure 3.17 Implemented FPGA Chip of Variable Precision Two-Dimensional Fine Grain Pipeline FIR Filter

3.7 SUMMARY

The variable precision two-dimensional fine-grain pipeline technique proposed in this work consumes less power than that of the existing methods. The variable precision two-dimensional fine-grain pipeline technique is applied to an 8-tap FIR filter. A comparison of this method with the existing techniques clearly shows that, the proposed one enables a power reduction of 18% with an area addition of only 3% in the FIR filter without affecting the speed and output quality.