CHAPTER 6

ASYNCHRONOUS QUASI DELAY INSENSITIVE TEMPLATES (QDI) BASED VITERBI DECODER

6.1 INTRODUCTION

Asynchronous designs are increasingly used to counter the disadvantages of synchronous designs. This chapter reveals the design of an asynchronous Viterbi decoder using QDI templates. The roadmap of the chapter is given as follows: Section 6.1 describes the advantage of asynchronous design, problems in the synchronous design, asynchronous channels and the QDI templates used in the design. Sections 6.2, 6.3 and 6.4 explain the asynchronous BMU, ACS and SMU with internal transistor level circuits. Section 6.5 informs the integrated design of asynchronous Viterbi decoder. At last, sections 6.6, 6.7 and 6.8 discuss the simulation results and performance comparison of the proposed work with the synchronous and existing literature survey.

The notable problems due to synchronous system designs are clock skew, power dissipation, interfacing difficulty and worst case performance. It is therefore not surprising that the area of asynchronous circuits and systems, which generally do not suffer from these problems, is experiencing a significant resurgence of interest in research activity. QDI design is a practical approximation to DI design. QDI circuit works correctly regardless of the delay of signal (William Benjamin Toms 2006) within the circuit.
6.1.1 Asynchronous Communication Channels

Asynchronous circuits are composed of blocks that communicate to each other using handshaking via asynchronous communication channels, in order to perform the necessary synchronization, communication, and sequencing of operations. Asynchronous communication channel consists of a bundle of wires and a protocol to communicate the data between the blocks. There are two types of encoding scheme for data handling in asynchronous channels. The single-rail encoding shown in Figure 6.1 uses one wire per bit to transmit the data and a request line to identify the validity of the data and the associated channel is called a bundled-data channel.

Alternatively, in dual-rail encoding as shown in Figure 6.2 the data is sent using two wires for each bit of information. Dual-rail encoding allows data validity to be indicated by the data itself. It is often used in QDI designs. Hence in the proposed asynchronous design of Viterbi decoder, the 4 phase handshaking protocol in dual rail encoding scheme is preferred. Compared to the 2 phase handshake protocol, the 4 phase protocol has less area overhead.

![Figure 6.1 Single Rail Encoding](image1)

![Figure 6.2 Dual Rail Encoding](image2)
The asynchronous design is based upon QDI templates like PCHB, WCHB, and the completion of the operation is ensured by a C-element. QDI templates prevent unnecessary transients and avoid delay in the circuits, thereby minimizing the power consumption.

### 6.1.1.1 Template of WCHB Buffer

WCHB template with a left (L) and right (R) channel is shown in Figure 6.3. L0 and L1, R0 and R1 identify the false and true dual rail inputs and outputs respectively. Lack and Rack are active-low acknowledgment signals. When the buffer is in reset condition, all the data lines are low. The acknowledgment lines, Lack and Rack are set to high. When data arrives by asserting one of the input rails to high, the corresponding C-element output goes to low value, lowering the left-side acknowledgment Lack.

![Figure 6.3 WCHB Template](image)

After the data is propagated to the outputs through one of the inverters, the right environment asserts Rack to low value, acknowledging that the data has been received. Once the input data resets, the template raises Lack and resets the output. Since the L and R channels cannot simultaneously hold two distinct data tokens, this circuit is said to be a half buffer or half slack ½. The WCHB buffer has a cycle time of 10 transitions, and it is significantly faster than buffers based on other QDI pipeline templates.
6.1.1.2 Template of PCHB QDI

The PCHB template is shown in Figure 6.4. F refers to the logic function implemented by the nMOS transistors. The test for validity and neutrality is checked using an input completion detector. The input (Left) Completion Detector is denoted as LCD and the (Right) output Completion Detector as RCD.

![Figure 6.4 PCHB Template](image)

The template generates only an acknowledgment signal Lack after all the inputs arrive and the output has been evaluated by the function F. Request or precharge signal is pc and the enable signal is en. In particular, the LCD and the RCD are combined using a C-element to generate the acknowledgment signal. The advantage of PCHB template is that it uses only two elementary transitions and has short latency when used in the design stages.
6.1.1.3 C - Element

C-element is used to implement a completion detection circuit for self-timed or delay insensitive circuits. Figure 6.5 shows a two-input Muller C-element, with two inputs a, b and one output c.

![Figure 6.5 Muller C- Element](image)

If a = b = 1 then c = 1 and if a = b = 0 then c = 0, otherwise the value of c remains unchanged. This can be generalized to an n-input C-element. The output of an n-input C-element is 1 if all the inputs are 1 and it is 0 if all inputs are 0. Otherwise, its value remains unchanged.

6.2 DESIGN OF ASYNCHRONOUS BMU USING QDI TEMPLATES

The asynchronous BMU is illustrated in Figure 6.6. The architecture of the BMU comprises PCHB XOR gate and a 3 bit counter. Literals a and b (their complements) are the inputs for the XOR gate with a C-element and the output of the XOR gate is given to the 3 bit counter. The output is buffered using WCHB so that the corresponding BM values are obtained without any delay. C-element ensures completion of operation between the transistors.
6.2.1 DCVS Based XOR Gate

DCVS is a form of CMOS logic which requires differential inputs and generates two outputs (true and complement). This logic finds its application in implementing the asynchronous technique protocols i.e. Request and Acknowledge signals. Figure 6.7 shows the circuit diagram of DCVS based XOR gate which is used in the BM design. When the request line en goes high, nMOS transistors evaluates the logic and the required (true or complement) output alone is sent to the next stage. The inputs for the XOR gate are a, b and their complements are ab, bb. While the Enable signal or request signal is en. Precharge signal is represented as xe. Once the logic is evaluated and the output data is ready for the next stage the completion signal by C-element is set high. The 3-bit asynchronous counter is designed using the T FF, which internally has 3-input NAND gates, AND gate and OR gate.
6.3 ASYNCHRONOUS QDI BASED ACS UNIT

It consists of adder, comparator and selector unit. The SPICE design of asynchronous ACS unit is represented in Figure A 3.1, vide Appendix 3.

The main purpose of asynchronous adder is to add the BM and PM value. Asynchronous adder can be designed using different structures, Such as ripple carry adder, carry look ahead adder and carry save adder etc. Among these parallel adders (Abdellatif Bellaouar et al. 1995) ripple carry adder has the smallest area and low power. Ripple carry adder generally requires (Michael Brandon Roth 2004) fewer transistors and less layout area than the other designs. Here the 4-bit asynchronous ripple carry PCHB full adder is constructed by rippling four 1-bit asynchronous full adders. Asynchronous 4-bit adder architecture from SPICE is illustrated in Figure 6.8. Inputs to the adder are a [0:4], b [0:4], carry c and their complements.
Figure 6.8 Asynchronous 4-Bit Adder Architecture (SPICE)

The internal transistor level diagram for the one bit full adder (sum) is given in Figure 6.9.
The asynchronous PCHB and DCVS logic based full adder (sum) transistor level design is presented. The operation of the adder is given as: \(a\) and \(b\) represents the 2 input signals, sum (carry) output signals are represented as: \(s_1\) (\(d_1\)) and \(s_0\) (\(d_0\)), \(en\) and \(se\) (\(de\)) are asynchronous PCHB logic handshaking signals. When the \(en\) and \(se\) signals are active low, the pMOS pull-up transistors are turned on and outputs \(s_0\), \(s_1\) obtain their logic values. The same operation is performed for the carry circuit.

### 6.3.1 Asynchronous 4-Bit Comparator

The output of the adder is fed to the next level i. e. comparator unit. The SPICE diagram shown in Figure 6.10 represent the 4-bit magnitude asynchronous comparator that consists of PCHB based XNOR gate, AND gate, OR gate and WCHB buffer.
Asynchronous AND gate is shown in Figure 6.11. Operation of the gate is explained as: en and ae are asynchronous PCHB logic handshaking signals. When the inputs a and b are high the two nMOS transistors connected in series generates an output 1 at a1. During the evaluation of the input signals the en and ae signals are in active high mode.
6.3.1.2  Asynchronous OR Gate

Asynchronous OR gate is shown in Figure 6.12. The operation of the gate is given as: en and oe are asynchronous PCHB logic handshaking signals. When the inputs $a=1$ and $b=0$ the two nMOS transistors connected in parallel generate a high output at $o_1$. 

![Figure 6.11 Asynchronous AND gate](image1)

![Figure 6.12 Asynchronous OR Gate](image2)
6.3.1.3  Asynchronous XNOR Gate

The asynchronous XNOR gate is shown in Figure 6.13. When \( a=1 \) and \( b=1 \), the two nMOS transistors \( a \) and \( b \) connected in series turn on and produce an output of one at \( x_1 \).

![Figure 6.13 Asynchronous XNOR Gate](image)

Thus the internal architecture of the comparator consists of the aforesaid gates that have been delineated so far.

6.3.1.4  Asynchronous 4-Bit Selector Unit

The selector is actually an asynchronous multiplexer. Figure 6.14 shows the SPICE diagram of 4-bit selector unit. After addition and comparison the selector outputs the minimum PM that is based on the decision of the comparator. The inputs to the selector units are \( a_{00}, a_{01}, b_{00}, b_{01},\ldots,a_{30}, a_{31}, b_{30}, b_{31} \). Among the two 4 bit inputs, the comparator produces the minimum value \( (a<b) \) as input to the select lines which is \( ss_0, ss_1 \).
6.4 DESIGN OF ASYNCHRONOUS SMU USING TRANSPARENT LATCHES

The comparator output is given to the select line of the MUX and the associated inputs are shifted to each register. Architecture of SMU for single stage is represented in Figure 6.15. In the architecture the inputs a, ab, b, bb are the inputs of the SMU unit and the configuration of the register is serial in serial out fashion. The asynchronous survivor memory unit consists of 2:1 multiplexer (selector unit) with asynchronous latches and to match the delay buffers are added.
Registers are constructed by means of asynchronous latches (Transparent latch). Data shift register is constructed by transition latches (Rostislav Dobkin et al. 2006). This asynchronous technique involves the structure of capture passes storage logic (Paul Day et al. 1995) for the design of latches. The advantage of the capture pass latch over the other latches such as double edge triggered d-latch, set-reset latch is that it avoids the unnecessary switching, has high performance and occupies minimum area.

The capture-pass latch is transparent until an event occurs on the capture line. This causes the latch to hold any data input Din that is on its input line. The capture done event indicates that the capture operation has been finished. Dout has the input value and further change in the input does not affect the output. An event on the pass signal makes the latch to go its transparent state and to ensure this operation was completed and an event on pass done signal is activated.

6.5 INTEGRATED DESIGN OF VITERBI DECODER USING PCHB AND WCHB TEMPLATES

Viterbi decoder comprises three blocks and in the proposed design the three stages are connected in a linear fashion using the WCHB and PCHB templates. The integrated design of asynchronous Viterbi decoder is presented in Figure 6.16.
The operation of the asynchronous design is explained with respect to a state transition graph. When the first data is given as input for the BMU, LCD1 generates a signal to turn on C1 in order to enable the pc and en signals. The given input data is evaluated by the BMU. When the outputs of BMU are validated, completion signal from the RCD1 is sent to the C1 of the BMU stage and LCD2 of the ACS stage. Now ACS unit starts evaluating the data. As soon as the output of ACS is valid, RCD2 generates a completion signal to C2 and acknowledgement signal to Lack in the BMU stage, also a request signal to LCD3 unit of SMU. Now BMU unit goes to the precharge phase and SMU is ready for evaluation of data. Thus the three stages execute in a linear pipeline fashion without pipelining registers. The control signals such as se, en, pc, Lo, L1, Ro, R1, and C are designed separately and the circuit is connected in the design wherever necessary.
6.6 PERFORMANCE METRICS COMPARISON WITH SYNCHRONOUS VITERBI DECODER

The performance comparison of the proposed asynchronous Viterbi decoder with the synchronous design and as well as the previously designed works are discussed as follows.

6.6.1 Synchronous Viterbi Decoder

Synchronous Viterbi decoder is designed in order to compare the performances with the asynchronous design. In synchronous design, a global clock is used to synchronize the operation. Internal modules of synchronous Viterbi decoder are designed using DCVS logic based transistor level circuits. They are represented in Appendix III. Figure A 3.2 shows the synchronous AND gate, Figure A 3.3 presents synchronous OR gate, synchronous 2:1 multiplexer is given in Figure A 3.4 and the synchronous XNOR gate is illustrated in Figure A 3.5.

6.7 SIMULATION RESULTS OF ASYNCHRONOUS QDI DESIGN

The Viterbi decoder is simulated in T-SPICE to obtain timing behavior and power consumption. For K>9 the complexity of the decoder increases.

6.7.1 Output Waveform of BMU

The output waveform of BMU using asynchronous QDI is illustrated in Figure 6.17.
Figure 6.17 Output of BMU Using Asynchronous QDI

The two dual rail inputs to the BMU are a(a, ab) and b(b, bb). The dual rail value x(x0, x1) be the output of the XOR gate, which becomes the input for the counter unit. The output values of the BMU are given by Q11,Q10,Q20,Q21,Q30 and Q31.

6.7.2 Simulation Results for ACS Unit

The complete output of ACS signals could not be viewed clearly in SPICE. Outputs of the three blocks adder, comparator and selector are given separately.

6.7.2.1 Output Waveform of Asynchronous 1- Bit Full Adder

The output waveform of a full adder is shown in Figure 6.18. The three dual rail inputs of single full adder in a ripple carry adder are a (a00, a01), b (b00, b01) and c (c0, c1). The dual rail outputs of full adder are sum s (s00, s01).
6.7.2.2 Output Waveform of Selector Unit

The output of selector unit is shown in Figure 6.19. The dual rail inputs of the selector unit for a single stage is (a00, a01) and b0 (b00, b01). The dual rail select inputs are ss0 and ss1. The dual rail outputs of the selector unit is s (s0, s1). When the signal value of the select input is Boolean logic ‘1’ then the value present in the a input is transferred to the output. When the select input is Boolean logic ‘0’ then the b input value is transferred to the output s0 and s1.
6.7.2.3 Output of Capture Pass Transparent Latch

The control signals of the single capture pass transparent latch are the $c$, $cd$, $p$, and $pd$. Signals $do$ and $d1$ represent the inputs. Figure 6.20 gives the output for the capture pass latch.
6.7.3 Complete Output of the Viterbi Decoder with Request and Acknowledge Signals

The output waveform of Viterbi decoder is given in Figure 6.21. The inputs to the single path of the decoder are given as a (expected sequence) and b (received sequence) and their complements. When the request signal is kept high, BMU is obtained by the hamming distance of the input values. Now a request signal from the BMU is sent to the ACS unit and the acknowledge signal is sent back to the BMU. Thus the minimum PM value is selected for the given input sequence. The dual rail output of the Viterbi decoder is \( VD_{out}^0 \) and \( VD_{out}^1 \). The block diagram of Viterbi decoder uses two BM units since each state have two branches in the trellis. Here the expected sequence is \( a = 11 \ 01 \ 11 \) and the received sequence for the first BM is \( b = 00 \ 10 \ 01 \) and the received sequence for the second BM is \( d = 11 \ 01 \ 10 \) and the decoded output sequence is \( VD_{out}^1 = 11 \ 0 \).

![Figure 6.21 Output Waveform of Viterbi Decoder](image)

(Required Signals Alone Represented)
6.8 RESULTS AND DISCUSSION

Basic building blocks of the Viterbi decoder are designed in both synchronous and asynchronous techniques using TSMC (Taiwan Semiconductor Manufacturing Company) in 0.25μm CMOS technology with 2.5V\textsubscript{dd}. Table 6.1 contains the performance comparison of synchronous and asynchronous design. The simulation result illustrates that the asynchronous circuit has a high transistor count with a frequency of 425MHz when compared to the synchronous circuit.

Table 6.1 Performance Comparison of Synchronous and Asynchronous Design

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No. of Transistors</td>
</tr>
<tr>
<td>Synchronous Design</td>
<td>9215</td>
</tr>
<tr>
<td>Asynchronous Design</td>
<td>16802</td>
</tr>
</tbody>
</table>

The power consumption of Viterbi decoder for various constraint lengths $K=3$ to 7 is shown in Table 6.2.
Table 6.2 Comparison of Power Consumption of Viterbi Decoder

<table>
<thead>
<tr>
<th>Constraint Length $K$</th>
<th>Proposed Synchronous Method</th>
<th>Existing 4 phase Single Rail Encoding Asynchronous Design</th>
<th>Proposed Asynchronous QDI Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>140.14</td>
<td>61.736</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>141.26</td>
<td>61.79</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>140.23</td>
<td>6 1.82</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>142</td>
<td>62.85</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>141.65</td>
<td>61.765</td>
<td></td>
</tr>
<tr>
<td>Average power Consumption</td>
<td>141.56mW @ 320MHz</td>
<td>85mW @ 426 MHz</td>
<td>61.99mW @ 425 MHz</td>
</tr>
</tbody>
</table>

Asynchronous design has 56.20% less power consumption when compared to synchronous design. It has 27% reduced amount of power consumption than the existing 4 phase protocol with single rail encoding (Mohamed Kawokgy et al. 2004) asynchronous design with almost the same frequency.

Table 6.3 shows the comparison of proposed asynchronous technique with the techniques from the literature survey. Javadi et al. (2003) showed the comparisons of various asynchronous techniques. The pictorial representation of power consumption of different logic styles is given in Figure 6.22.
Table 6.3  Comparison of Viterbi Decoder Designs from Literature Survey

<table>
<thead>
<tr>
<th>Design</th>
<th>Technology</th>
<th>$V_{dd}$ (v)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronous (reference)</td>
<td>0.35µm</td>
<td>n/a</td>
<td>203</td>
</tr>
<tr>
<td>Systolic array</td>
<td>0.5µm</td>
<td>3.3</td>
<td>280</td>
</tr>
<tr>
<td>SPL</td>
<td>0.35µm</td>
<td>2.5</td>
<td>88</td>
</tr>
<tr>
<td>Self timed</td>
<td>0.35µm</td>
<td>n/a</td>
<td>1333</td>
</tr>
<tr>
<td>Asynchronous QDI[Javadi]</td>
<td>0.35µm</td>
<td>3.3</td>
<td>166</td>
</tr>
<tr>
<td>Asynchronous QDI[Javadi]</td>
<td>0.35µm</td>
<td>2.5</td>
<td>85</td>
</tr>
<tr>
<td>Optimized ACS</td>
<td>0.35µm</td>
<td>3.3</td>
<td>109</td>
</tr>
<tr>
<td>Optimized ACS</td>
<td>0.35µm</td>
<td>2.5</td>
<td>62</td>
</tr>
<tr>
<td>Proposed Asynchronous PCHB &amp; DCVS design</td>
<td>0.25µm</td>
<td>2.5</td>
<td>61.9</td>
</tr>
</tbody>
</table>

Figure 6.22 Comparison of Power Consumption of Different Logic Styles
The proposed asynchronous method featured a power reduction from 4.6% to 72.9% with that of the existing asynchronous methods. A comparison of the proposed asynchronous QDI method with the synchronous method in parameters like speed and delay reveals that it has 1.32 times improvement in speed with a reduced delay of 2.13ns.

**Table 6.4 Comparison of Power consumption for Architecture K=3 and K=7**

<table>
<thead>
<tr>
<th>Constraint Length K</th>
<th>Proposed Asynchronous QDI Method (With architecture K=3)</th>
<th>Proposed Asynchronous QDI Method (With architecture K=7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>61.736</td>
<td>89.32</td>
</tr>
<tr>
<td>4</td>
<td>61.79</td>
<td>89.65</td>
</tr>
<tr>
<td>5</td>
<td>61.82</td>
<td>89.87</td>
</tr>
<tr>
<td>6</td>
<td>62.85</td>
<td>89.67</td>
</tr>
<tr>
<td>7</td>
<td>61.765</td>
<td>89.74</td>
</tr>
<tr>
<td>Average power</td>
<td>61.99mW</td>
<td>89.65mW</td>
</tr>
</tbody>
</table>

In the proposed four methods, for K=3 the Viterbi decoder architecture has slightly increasing power consumption for different constraint length inputs. In order to validate this, simulation is carried out for the asynchronous circuit method with K=7 for 64 state which reveals there is an increase in power consumption from the previous lengths K < 7. From Table
6.4 the deviation of power consumption for \( K=3 \) and \( K=7 \) is 27mW. The expected power level of this method for \( K=7 \) is 89.65mW. It also depends on the most favourable choice of constraint length for different applications. Simulation result for \( K=7 \) is carried out for the Viterbi decoder and the output waveform for \( K=7 \) is presented in Figure A 3.2 in Appendix 3.

It should be explained whether power reduction benefits will be still valid for soft decision Viterbi decoder implementation. What is the expectation for the improvement in power?

The proposed methods are experimented to hard decision Viterbi decoder, but based on the existing literature survey the methods can also be equivalently applied for soft decision decoding which can result in optimum power consumption.

6.9 CONCLUSION

The asynchronous design of Viterbi Decoder using QDI templates is an alternate style which avoids global clocks. In this regard, the design of circuits using PCHB, WCHB, and DCVSL is presented clearly. The asynchronous design was based upon QDI timing model implemented in DCVSL which can be used for robust and low power applications. The simulation results proved that the asynchronous design has less power consumption when compared to synchronous design.