

LOW POWER VLSI ARCHITECTURE OF A VITERBI DECODER USING ASYNCHRONOUS PRECHARGE HALF BUFFER DUAL RAILTECHNIQUES

T.Kalavathidevi¹ C.Venkatesh²

¹Faculty of Electrical Engineering, Kongu Engineering College, Erode, India.
kalavathidevi@gmail.com

²Professor, Department of ECE, Surya Engineering College, Erode, India.
cvkongu@gmail.com

Abstract

In modern communication systems, reducing power consumption has become a fundamental design goal, especially for VLSI integrated circuits used in mobile communication systems. Asynchronous design is progressively becoming more an attractive alternative to synchronous design because of its potential for high-speed and low-power. Asynchronous circuits will therefore generally dissipate less power than synchronous ones. The proposed method is focused on the design of VLSI architecture for a Viterbi Decoder by using low power VLSI design techniques at circuit level with asynchronous self timed control and Differential Cascade Voltage Switch Logic (DCVSL). The asynchronous designs based on Pre Charged Half buffer (PCHB) templates. The asynchronous Viterbi decoder comprises of BMU, PMU and a SMU. Communication within the decoder blocks are controlled by the Request-Acknowledge handshake pair which respectively signals that valid data is ready and that it has been accepted. The design of various units of Viterbi Decoder is done by using T – SPICE in TSMC 0.25um technology. The simulation results shows 90% power reduction has been achieved by using asynchronous design technique compared to that of the synchronous design.

Keywords: Viterbi Algorithm, Viterbi decoder, Asynchronous, DCVSL, QDI templates, PCHB, Low Power, T – SPICE

1 INTRODUCTION

The viterbi algorithm [3, 4] is used widely in the digital transmission field as a means of decoding convolution forward error correction codes. The Viterbi algorithm is a maximum likelihood (ML) probability method; the output of the decoder is the sequence most likely to have been transmitted, given an observed received sequence from an encoder. Since the received signal is Analog, it can be quantized into several levels. If the received signal is converted into two levels, either zero or one, it is called hard decision. Asynchronous design is becoming an increasingly attractive alternate to synchronous design as they assume binary signals, but there is no common and discrete time. Instead the circuits use handshaking between their components in order to perform the necessary synchronization, communication, and sequencing of operations. This difference gives asynchronous circuit's inherent properties [5] that can be exploited to advantage in the areas such as Low power consumption, High operating speed, less emission of electro-magnetic noise and no clock distribution and clock skew problems.

This paper presents a low power asynchronous VLSI architecture for Viterbi decoder to reduce power dissipation with increased speed; this can be achieved by adopting differential cascade voltage switch logic. The architecture is implemented in both synchronous and Asynchronous techniques. The latter design is simpler and more efficient.

This paper is organized as follows. Section I introduction about the viterbi algorithm and encoder. Section II describes the background on asynchronous design and viterbi decoders. Section III Asynchronous channels. Section IV explains the architectural design of decoder Section V presents the Dual rail logic implementation techniques and Section VI discuss the experimental results and comparison.

The basic building blocks of the encoder are a convolution and parity calculation. The last k bits of the input stream are stored in a shift register and these are convolved (XORed) with a k -bit long pattern and the parity of the resulting stream is the encoders output.. The ratio of input bits to output bits (k/n) called the code rate, which can be selected based on the design. The designed decoder is a $\frac{1}{2}$ -rate with a constraint length of 3. The encoder in Figure 1.1 produces two bits of encoded information; it is called as $\frac{1}{2}$ rate encoder. The convolution encoder is a mealy machine, where the output is a function of the current state and the current input. The stream of information bits flows in to shift register from one end and is shifted out at the other end. Location of the stages is determined by the interconnection function. The location of the stages as well as the number of memory elements determines the minimum hamming distance, intermediate states has been stored by positive edge triggered D-Flip Flop.

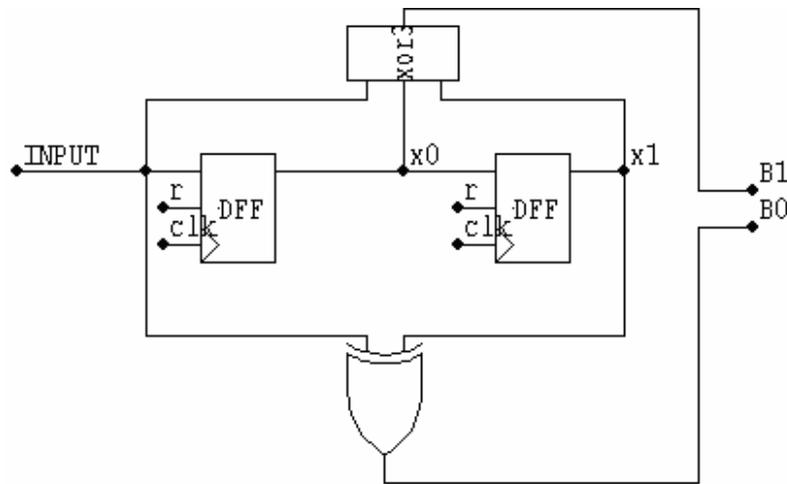


Figure 1.1 The Convolutional Decoder with $r=1/2, K=3$

Operation of the convolutional encoder [9] can be explained in a state diagram as in figure 1.2, as there are two memory elements in the encoder, there are four possible states, these states are named as S_0 to S_3 . Each input to a state generates an encoded output code and causes a transition. For each state, there are two outgoing transitions-one corresponding to a 0 input bit and the other corresponds to 1 input bit. Initially when the current state is at S_0 , if the input is 1 the state S_0 moves to the state S_1 . In this paper, Maximum Likelihood algorithm with hard decision has been designed.

An ML path is found with the aid of a Branch metric and a Path metric. A Branch metric is the Hamming distance between the estimate and the received code symbol. The branch metrics accumulated along a path form a path metric. A partial path metric at a state, often referred as state metric, is the path metric for the path from the initial state to the given state. After the trellis grows to its maximal size, there are two incoming branches for each node. Between two branches, the branch with a smaller (in terms of Hamming distance) partial metric survives, and the other one is discarded. After surviving branches at all nodes in the trellis have been identified, there exists a unique path starting and ending at the same initial state in the trellis. The decoder generates an output sequence corresponding to the input sequence for this unique path.

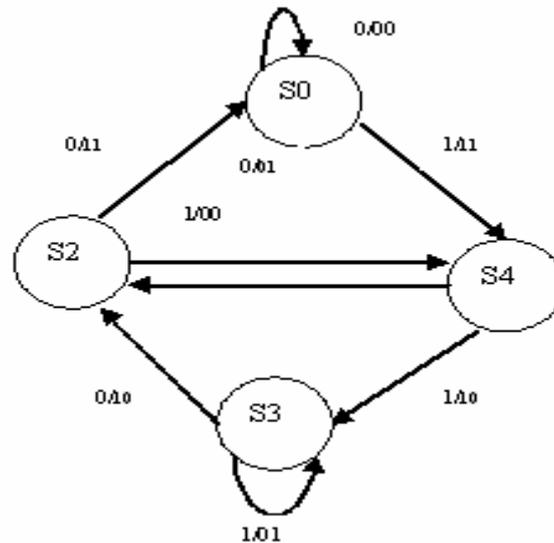


Figure 1.2 State Diagram of convolutional encoder

The asynchronous design is based upon Quasi Delay Insensitive (QDI) [7]: Quasi delay insensitive design is a practical approximation to delay insensitive design. QDI circuit work correctly regardless of the delay of signal with in the circuit, except for some assumptions about certain delay. The QDI assumption has also been extended to include assumptions of isochronic propagation through a number of logic gates. Commonly used QDI templates are the Weak-Conditioned Half Buffer (WCHB), the Precharged Half Buffer (PCHB), and the Precharged Full Buffer (PCFB) templates. In this paper we have used the QDI templates called Pre-Charged Half Buffer (PCHB) and Weak Conditioned Half Buffer (WCHB).

2 BACKGROUND

This section describes background on viterbi decoder and on Caltech's WCHB and PCHB templates.

The Viterbi algorithm (VA)[4,9], widely used in digital communication, is known to be an efficient method for the realization of maximum likelihood decoding of convolutional codes.

Based on the modified T-algorithm VLSI architecture for the VD with reduced computations in add compare select unit is developed.[10] In this paper, an architecture of the VD and its corresponding VLSI implementation is developed for decreasing the average power dissipation. Several attempts for increasing the throughput and reducing the power dissipation and the area in the ACS have been reported in the literature.

The architectures for Add-Compare-Select (ACS) unit [2] In this a new VLSI architecture is proposed for carrying out the Add-Compare-Select (ACS) operation for the Viterbi decoder which can reduce the complexity of the computation.

A novel pre-computational architecture is proposed to further reduce the power consumption of the ACS unit. Examples in the second group include SPL [1] (Single ended Pass transistor Logic) implementation shows that dynamic power dissipation in ACS parts of the Viterbi decoder is reduced.

The paper [6] presents an area efficient, low-power and robust ACS unit for Viterbi Decoder in two synchronous and asynchronous architectures. The architecture uses a hybrid CMOS-Pseudo NMOS technology to improve area and throughput factors. They concentrated only in ACS unit and not the full decoder. The asynchronous QDI template used was PCFB.

The paper [8] describes a design for a self timed Viterbi decoder. The new design is based upon serial, unary arithmetic for the manipulation and storage of metrics. The new architecture

occupies between 29% and 23% less area than a selection of synchronous implementations with the same design parameters which use the same process and cell-library.

The asynchronous design methodology [7] proposed by Caltech is perhaps the most robust using delay-insensitive communication between quasi delay insensitive (QDI) pipeline templates.

2.1 SYNCHRONOUS AND ASYNCHRONOUS TECHNIQUES

Digital VLSI circuit design styles can be mainly classified as either synchronous, asynchronous or some mixture. Most digital circuits designed and fabricated today are “synchronous”. In essence, they are based on two fundamental assumptions that greatly simplify their design: (1) all signals are binary, and (2) all components share a common and discrete notion of time, as defined by a clock signal distributed throughout the circuit.

Synchronous designs, consists of subsystems, which are controlled by one or more clocks that control synchronization and communication between blocks. Combinational logic is placed in between clocked registers that hold the data. The delay through the combinational logic plus relevant setup time should be smaller than the clock cycle time. In fact, the data at the inputs of the registers may exhibit glitches or hazards as long as they are guaranteed to settle before the sample clock edge arrives.

Asynchronous circuits are fundamentally different; they also assume binary signals, but there is no common and discrete time. Instead the circuits use handshaking between their components in order to perform the necessary synchronization, communication, and sequencing of operations. Asynchronous methodologies, in contrast, use event-based handshaking to control synchronization and communication between blocks. Micro pipelines (Asynchronous pipeline)[6,7,8] belong to an important asynchronous architecture. The two modules used instead of clocking strategies are the Weak Charge Half Buffer (WCHB) and Pre Charge Half Buffer (PCHB).

2.2 ASYNCHRONOUS CHANNELS

In general, asynchronous designs are composed of blocks communicating using handshaking via asynchronous communication channels [7]. An asynchronous communication channel is a bundle of wires and a protocol to communicate data between blocks. The encoding scheme in which one wire per bit is used to transmit the data and an associated request line is sent to identify when data is valid is called single-rail encoding and is shown in Figure.2.1a. The associated channel is called a bundled-data channel. Alternatively, in dual-rail encoding in figure2.1b the data is sent using two wires for each bit of information. Both single-rail and dual-rail encoding schemes are commonly used, and there are tradeoffs between each. Dual-rail and 1-of-N encoding allow for data validity to be indicated by the data itself and are often used in QDI designs.

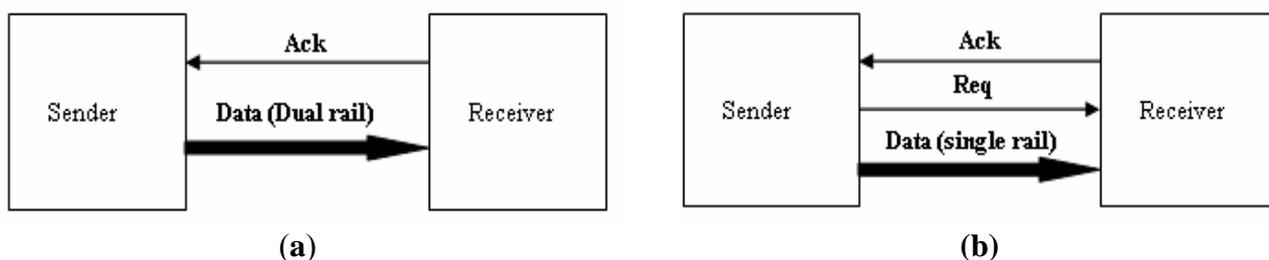


Figure 2 1. Pipeline channels (a) Bundled data channel. (b) Dual-rail channel

2.3 WEAK CONDITIONED HALF BUFFER (WCHB)

Figure 2.2 shows a WCHB template [5] for a linear pipeline with a left (L) and right (R) channel and an optimized WCHB dual-rail buffer. 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.

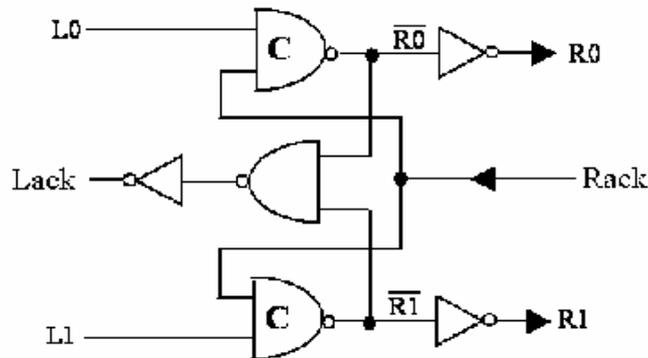


Figure 2.2 Weak Conditioned Half Buffer

The operation of the buffer is as follows. After the buffer has been reset, all data lines are low and acknowledgment lines, Lack and Rack, are high. When data arrives by one of the input rails going high, the corresponding C-element output will go low, lowering the left-side acknowledgment Lack. After the data is propagated to the outputs through one of the inverters, the right environment will assert Rack low, 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 has slack $\frac{1}{2}$. This WCHB buffer has a cycle time of 10 transitions, which is significantly faster than buffers based on other QDI pipeline templates.

2.4 PRE CHARGED HALF BUFFER (PCHB)

Figure 2.3 shows the template for a pre-charged half-buffer (PCHB)[7]. The test for validity and neutrality is checked using an input completion detector. The Input Completion Detector is denoted as LCD and the Output Completion Detector as RCD.

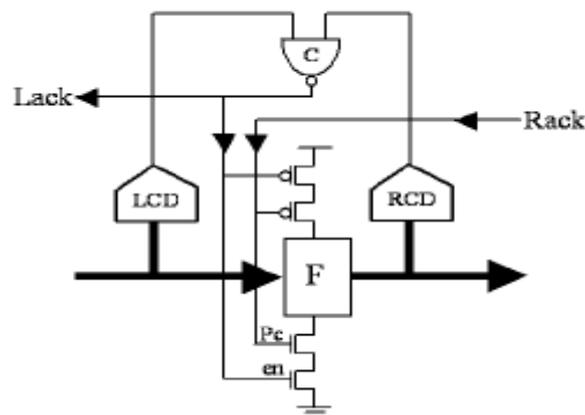


Figure 2.3 Pre Charged Half Buffer

The function block need not be weak-conditioned logic and thus can evaluate before all the inputs have arrived (if the logic allows). However, the template only generates an acknowledgment signal Lack after all the inputs have arrived and the output has evaluated. In particular, the LCD and

the RCD are combined using a C-element to generate the acknowledgment signal. In addition to yielding delay-insensitive (DI) communication between cells, the PCHB is internally quasi delay insensitive, meaning that it has no significant timing assumptions. As the C-element is inverting the acknowledgment signal is an active-low signal and the Lack signal is often buffered using two inverters before being sent out. Another two inverters are also often added to buffer the internal signal *en* that controls the function block

2.5 MÜLLER C – ELEMENT

The Muller C-element [11] is indeed a fundamental component that is extensively used in asynchronous circuits is used to implement a completion detection circuit for self-timed or delay insensitive circuits. . It is a state-holding element much like an asynchronous set-reset latch Figure 2.4 shows a two-input C-element, with two inputs a, b and one output c.

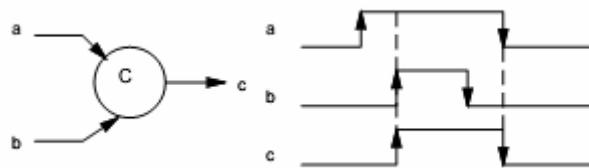


Figure 2.4 Muller C-element

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 inputs are 1 and it is 0 if all inputs are 0; otherwise its value remains unchanged.

3. ASYNCHROUS DESIGN IN VITERBI DECODER

Figure 3.1 shows the block diagram of Asynchronous Viterbi decoder. This saves power as it is not required to generate or distribute a global clock. Instead, timing between blocks is performed by local handshake signals. The clock signal is replaced by some form of handshaking between neighbouring registers; for example the simple request-acknowledge based hand shake protocol.

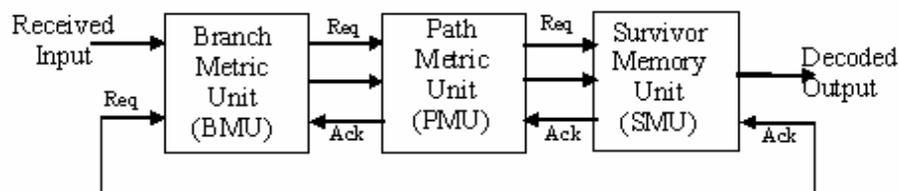


Figure 3.1 Asynchronous Viterbi Decoder

An asynchronous approach encourages designers to design in a modular manner. This means that sections can operate independently, concurrently and at their fastest natural rate that is handshake signals connects one register to the next. The handshaking between registers; a combinatorial circuit simply absorbs a token on each of its input links, performs its computation, and then emits a token on each of its output links. In the Viterbi decoder, the design partitions into two main operating sections namely the branch metric unit and the path back trace unit. These can operate asynchronously to each other and while the branch metric unit is linked to the input data rate, the back trace unit can be retracing at a higher rate which is clearly advantageous. In the asynchronous viterbi decoder received input from the encoder is given to the branch metric unit ,it calculates the hamming distance between the codes and at the request acknowledge signal from the path metric unit , minimum path is obtained which is stored in the survivor memory unit. At each

stage, request and acknowledge signals are provided in order to ensure the completion of the operation.

3.1 BRANCH METRIC UNIT (BMU)

The first step in the Viterbi decoding algorithm requires calculation of the branch-metric. The branch metric is the distance from the received code word to all the possible branch words. An implementation of the block is shown in Figure 3.2. The architecture comprises of a Xor gate and a counter. The branch word depends on the constraint length, the generator matrix, and the code rate. One input to the Xor gate is the received code symbol which is the encoder output and the other input is the generated sequence in the polynomial. Xor gate determines the difference in the number of transitions in the inputs and counter counts the total number of differing bits. In this case, the possible branch words are 00, 01, 10 and 11.

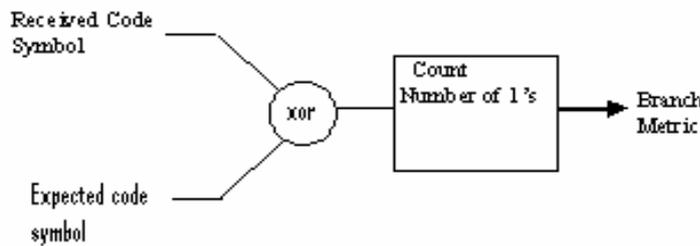


Figure 3.2 Branch Metric Computation Block

The hardware realization of the BMU computation block is shown in figure 3.3.

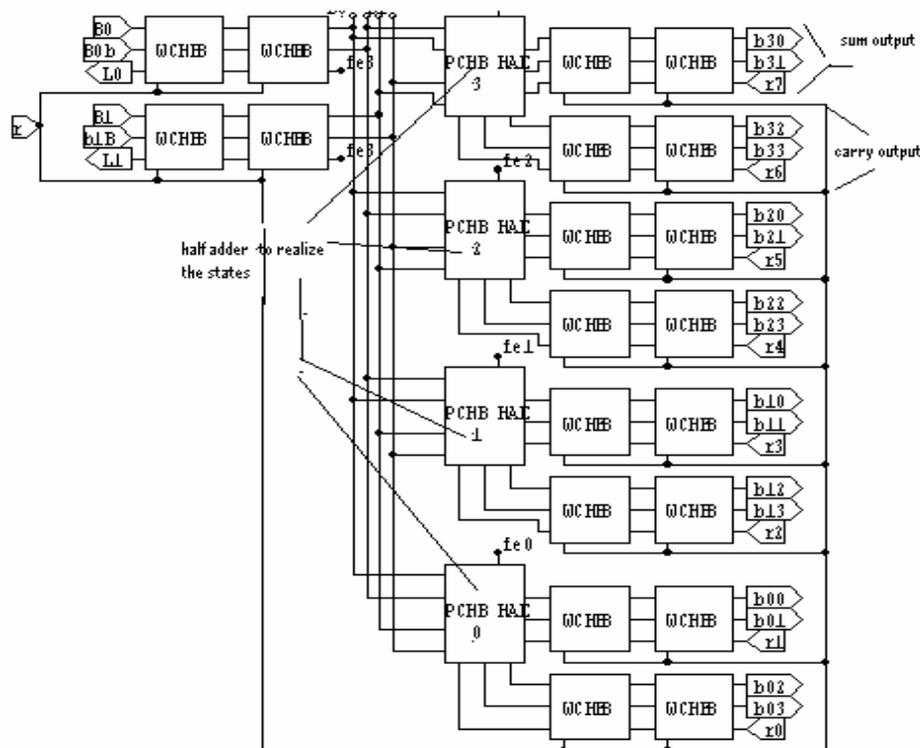


Figure 3.3 Hardware realization of Branch Metric Computation Block

The realization of the state diagram finds that it is the half adder, there are four states and each state has a path in the trellis hence for path a PCHB and WCHB buffer is designed using DCVSL. The outputs of the encoder and the expected code symbol from the trellis are taken as b_0 and b_1 inputs to the WCHB with a C-element, for each state a PCHB half adder is designed and the output is buffered using WCHB so that the corresponding four states are obtained as sum and carry outputs.

3.2 PATH METRIC UNIT (PMU)

The Path Metric Unit (PMU) calculates the new path metric values and decision values. Because each stage can be achieved from two states of earlier stage, there are two possible path metrics coming to the current state. The PMU adds the branch metric to path metrics and typically selects the smaller ones and makes a decision. The PMU stores the result of the addition as path metric for the current state. As the current state can be obtained from the earlier stage, the decision value can be represented as one bit. If the bit is one the path metric selected is from the lower state within the two possible states in the trellis diagram and if the bit is zero the path metric is selected from the upper state. The ACS (Add Compare Select) unit, which is the heart of the process and dictates the performance of the decoder. The ACS operation for each new state in the trellis performs the addition, comparison, and selection of the smallest path metric. Figure 3.4 shows the circuit diagrams of Asynchronous add compare select unit.

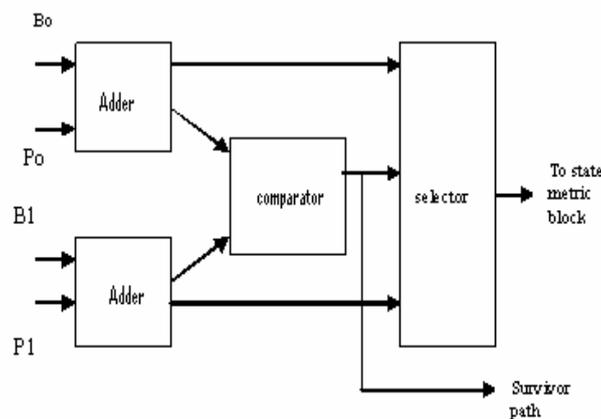


Figure 3.4 Add Compare Select Unit

Hardware realization of ACS unit in figure 3.5 has adder, comparator and selector. Two inputs b_0 and b_1 are given to the one of the two inputs of the adders. Initial value of the other input of the adder is taken as zero. In this method a 4 bit asynchronous ripple carry PCHB is constructed by rippling four 1-bit asynchronous full adders. The lower bits to the adder are the path metrics (previous branch metrics). A comparator then compares the resulting path metrics, and the lesser one is the output from the ACS unit. To exactly duplicate the trellis diagram and provide feedback, we need to place registers (buffers) at the outputs. This output is carried out to the comparator which also generates hand shake signal at the end of completion of comparison. After addition and comparison the selector outputs the minimum path metric that is based on the decision of the comparator.

To directly translate the state diagram into hardware, each node needs two 4-bit adders. The circuits are implemented by using DCVSL logic. For the constraint length of $K=3$ and code rate- $1/2$, it has four states and it requires 4-bit adder, comparator and selector. The four-state ACS unit updates path metrics for a single iteration of the trellis is shown in figure 3.5. Each unit consists of four two-way ACS units.

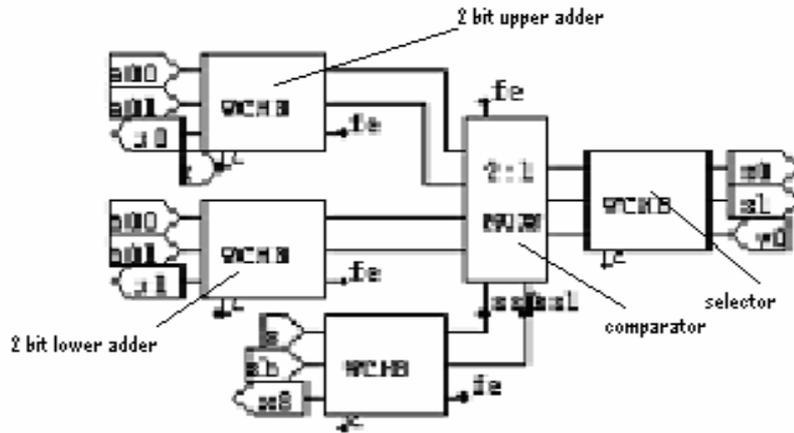


Figure 3.5 Hardware Realization of Add Compare Select Unit for 2 bit

3.3 SURVIVOR MEMORY UNIT (SMU)

The survivor path memory circuit includes survivor paths and decision bit paths. In this paper register Exchange method is implemented.

3.3.1 Register exchange (RE) method

In the RE approach, a register is assigned to each state. The register records the decoded output sequence along the path from the initial state to the final state. At the last stage, the decoded output sequence is the one that is stored in the survivor path register, the register assigned to the state with the minimum PM. In the architecture the inputs i_0, i_1, i_2, i_3 are the four outputs of the ACS unit and f_1, f_0 are the signals from the comparator output of ACS; the configuration of the registers is Serial In Serial Out. Since the RE method does not need tracing back, it is faster. The synchronous memory unit shown in figure 3.6 is constructed by using positive edge triggered FF and WCHB and DCVS logic.

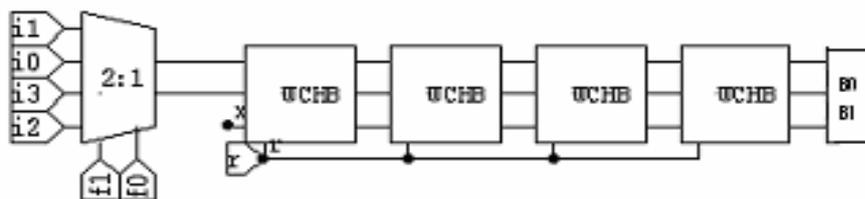


Figure 3.6 Survival Memory Unit

4 BUILDING BLOCKS OF THE INTERNAL ARCHITECTURES

The low power VLSI techniques used for the design at circuit level is the DCVSL logic. Differential Cascaded voltage switch logic (DCVS) is a dual rail CMOS technique in figure 4.1 which is a precharge logic that generates proper completion of true and complemented outputs also has the advantage over single rail traditional logic techniques in terms of Power dissipation, delay, fast switching, smaller load capacitance and high performance. In dual rail design the precharge P-type transistors may be replaced by transistor networks that detect when all the inputs are empty. The pull down networks will conduct only when the data paths are valid.

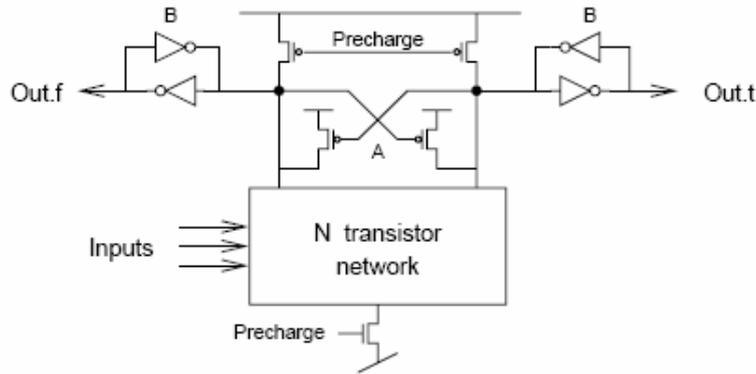


Figure 4.1 Precharge Dual rail logic CMOS networks

4.1 PCHB and DCVS FULL ADDER

The following figures show the DCVSL Arithmetic circuits along with QDI templates used in the architectures of BMU, PMU and SMU.

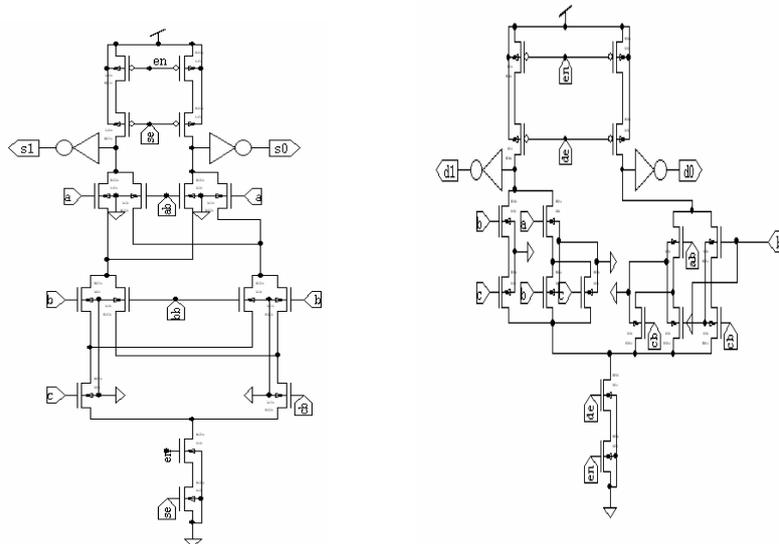


Figure 4.2 Full Adder sum and carry

The Figure 4.2 shows the asynchronous PCHB and DCVS logic based full adder sum and carry transistor level design. The operation of the adder is where a and b represent the 2 input signals, and s1 (d1) and s0 (d0) represents the true and complement sum (carry) output signal, en and se (de) are asynchronous PCHB logic handshaking signals. When the en and se signals are active low, the PMOS pull-up transistors turned on and s0, s1 are set to logic high. When the en, se signals become active high, the PMOS transistors turned off and depending on the input signals, either s0 (d0) or s1 (d1) is pulled to ground. The same operation is performed for carry.

4.2 PCHB and DCVS MULTIPLEXER

The transistor level diagram in Figure 4.34 shows the circuit level design of multiplexer mainly in selector of ACS Unit and SMU. a and b are the inputs to the multiplexer ,selection lines are s0,s1;based on the control signal from the comparator a or b is selected as the output.

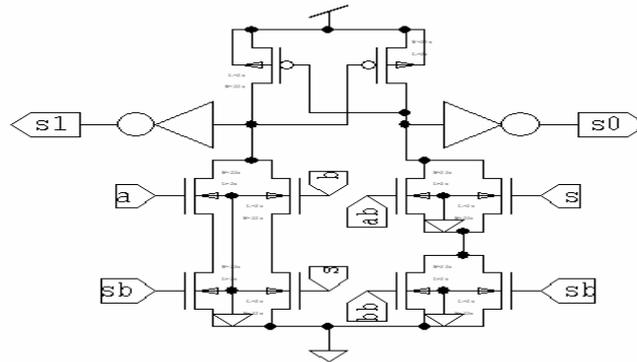


Figure 4.3 Multiplexer

5 EXPERIMENTAL RESULTS AND DISCUSSION

The functionality of the Viterbi Decoder is simulated using T-SPICE at TSMC 0.25 μm CMOS Technology. The input to the decoder is given by the output of the encoder in figure 5.1 with a sequence of 10011. First step execution is the result of branch metric unit obtained by the hamming distance, as per the design it is the four states, as in figure 5.2. Then the path metric value is calculated, added, compared and selected based on the input sequence. The final decoded value is 011011, For convenience only the true output of the logic is shown in figure 5.3. Request and Acknowledge signals are also represented in the figure 5.3. The synchronous circuit has the lower transistor count with reasonable power consumption. The asynchronous QDI has less power consumption with high transistor count.

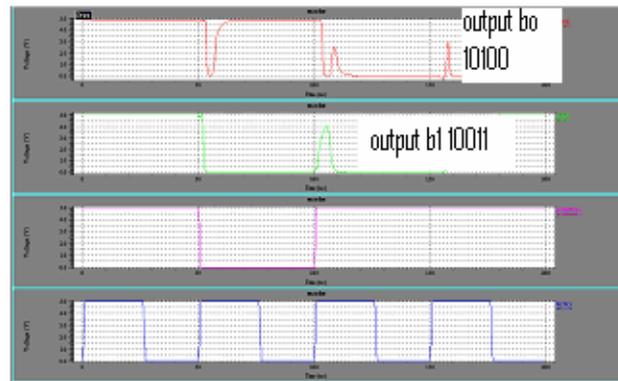


Figure 5.1 Output of Convolutional Encoder (B110011)

For every single input $K=1$, the shift register shift the content of the inputs and the output of the encoder is 2 bit. The encoder output b1 is 10011 or the input sequence 1 0011. The output of the encoder is the input to the Branch metric unit depending upon the inputs the sequence will follow the upper or lower path of the code tree.

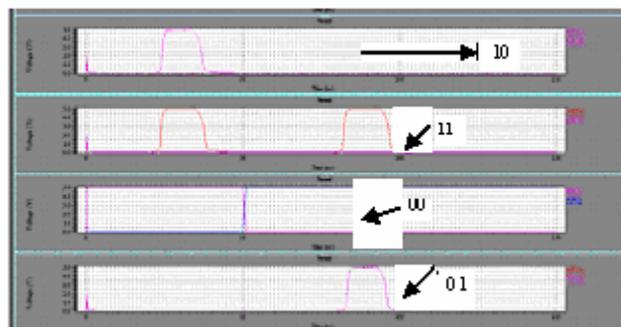


Figure 5.2 Output of Branch Metric Unit

The functionality of the Branch Metric Unit in figure 6.2 of the decoder is simulated and verified using TSMC 0.25µm CMOS technology in T-Spice with a supply of 2.5V. The outputs are the four states that is 10,11,01,00.

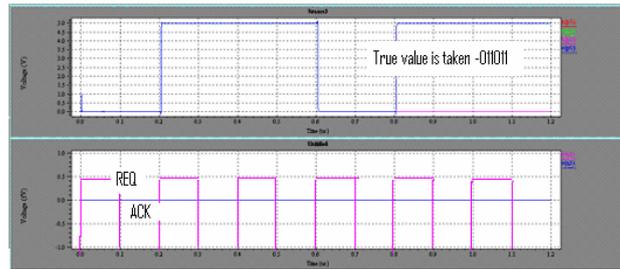


Figure 5.3 Output of Decoder (011011) taken only the true value of dual rail logic

The basic building blocks of the viterbi decoder are designed in both synchronous and asynchronous techniques using TSMC in 0.25µm CMOS technology. The results of only the true logic of dual rail are alone shown in the figure 6.3. Tanner design tool is used to design the basic building blocks and the simulation result shows that circuit runs at 425 Mbits/sec and consumes 1.73mW compared to the synchronous design simulated at TSMC 0.25 µm CMOS technology which consumes an average power of 20.4W with a a speed of 32Mhz.

5.1 PERFORMANCE COMPARISON OF SYNCHRONOUS AND ASYNCHRONOUS ARCHITECTURES

Synchronous and Asynchronous blocks of viterbi decoder are designed in 0.25-µm TSMC process with a 2.5V power supply. Comparison results of Synchronous and Asynchronous Design is discussed in this paragraph.

5.1.1 Branch Metric Unit :

T SPICE simulation of the BMU in table 5.1 shows a comparison of asynchronous QDI templates has transistor count of 806 with an average power consumption of 24mW when compared with the synchronous BMU which has a transistor count of 583 with a power consumption of 31mW.

Table 5.1 Comparison of Synchronous and Asynchronous BMU

Parameters	Synchronous	Asynchronous
Transistor	583	806
Power Consumption (W)	24mW	31mW
Speed	32MHz	425MHz

5.1.2 Add Compare Select Unit

Asynchronous DCVS logic based Adder compare select Unit or PMU simulated at 0.25 µm CMOS technology simulated using T-spice and results in table 5.2 shows the transistor count of 3810 with a power dissipation of 91mW and the synchronous design has a transistor count of 1834 with a power dissipation of 140mW..

Table 5.2 comparison of synchronous and Asynchronous ACS

Parameters	Synchronous	Asynchronous
Transistor	1834	3810
Power Consumption (W)	140 m	91 mw
Speed	32MHz	425MHz

5.1.3.Survivor Memory Unit

The SMU unit is also designed in both synchronous and Asynchronous techniques which has a performance of 5% increase in power reduction with a 3% increase in area with that of synchronous design. Asynchronous SMU unit has a power consumption of 8ns with that of synchronous has a power dissipation of 13ns

5.1.4 Viterbi Decoder

The internal blocks are designed and integrated to obtain the overall performance of the viterbi decoder in TSMC 0.25 μm CMOS technology and Table 5.3 shows the comparison of synchronous and Asynchronous design of viterbi decoder. The result shows that circuit has a speed of 425Mbits/sec and consumes 1.73mW of average power with a transistor count of 15702 compared to the synchronous design which consumes an average power of 20.4mW with transistors of 9215.

Table 5.3 comparison of synchronous and Asynchronous Viterbi decoder

Parameters	Synchronous	Asynchronous
Transistor	9215	15702
Power Consumption (W)	20.4m	1.73m
Speed	32MHz	425MHz

5.1.5 Analysis of BER Vs Eb/No

The BER in Figure 5.4 (the no. of errors when the decoded sequence is compared with the original stream as a function of Signal to Noise Eb/No in db) is obtained to validate the proposed Asynchronous Precharge half buffer based viterbi decoder in terms of decoding performance. The signal to noise ratio has determined the performance of the BER. In the conventional method of design [5] for a constraint length of $K=3$,code rate $\frac{1}{2}$ and frequency of 6db the BER is 6.81×10^{-13} and the proposed method has an BER of 4.45×10^{-30} . Analysis is performed for 3, 4, 5, 6, 7db. On comparison it indicates the proposed asynchronous architecture has a good BER performance with fast decoding capability with reduced power consumption.

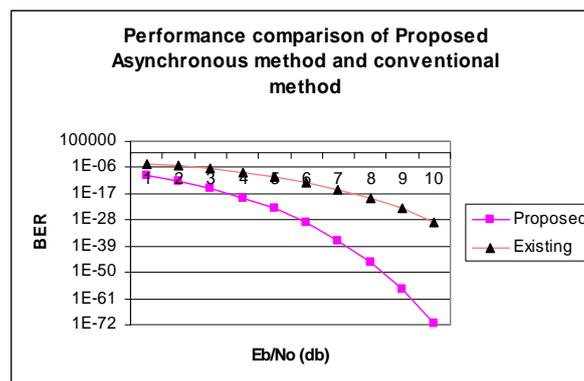


Figure 5.4 comparison of BER Vs Eb/No of the Viterbi decoder

The above figure shows the decoder has 54 % increased performance for detecting and correcting errors than the existing design [5] based on CMOS Pseudo logic. The overall decoding rate is 425Mbps.

6. CONCLUSION

Viterbi decoders employed in digital mobile communications are complex in its implementation and dissipate large power. The proposed Viterbi decoder uses asynchronous design

techniques to reduce power consumption. The asynchronous design was based upon Quasi Delay Insensitive (QDI) timing model which can be used for robust and low power applications. The asynchronous circuit design uses DCVSL logic. The simulation results show the asynchronous design has the decrease in power consumption by 90% with increase in transistor count by 0.7 times in relative to synchronous Viterbi decoder with code rate of $\frac{1}{2}$ and constraint length of $K=3$ in TSMC 0.25 μ m CMOS technology with a power supply of 2.5V. The performance of the asynchronous DCVSL logic based design shows reduced power consumption of 90% over the synchronous and 52 % of the existing method [5] based on CMOS-Pseudo techniques.

REFERENCES

1. Bogdan.I, Mumunteanu.M, Ivey.P.A, Seed.N.L, and Powell.N (2000) 'Power Reduction Techniques for a Viterbi Decoder Implementation', ESPLD 2000 (European Low Power Initiative for Electronic System Design) Third International Workshop, Rapallo, Italy, ISBN 90-5326-036-6, pp 28-48, July.
2. Chi-Ying Tsui, Cheng.R.S, K.Ling (1999) 'Low power ACS unit design for the Viterbi decoder', Proceedings of the IEEE symposium on Circuits and Systems, pp. 137-140.
3. Fettweis. G and Meyr. H (1991) 'High-speed parallel Viterbi decoding: algorithm and VLSI-architecture', IEEE Communications Magazine, 46v-55.
4. Forney. G (1973) 'The Viterbi algorithm', Proceedings of the IEEE, vol. 61, no.3, pp.268-278.
5. Jens Sparso, (2006), "Asynchronous Circuit Design A Tutorial", Technical University of Denmark
6. Mohammad K.Akbari, Ali Jahanian, Mohsen Naderi, Bahman Javadi (2004) 'Area Efficient, Low Power and Robust Design for Add-Compare-Select Units', Proceedings of the EUROMICRO Systems on Digital System Design (DSD'04) 0-7695-2203-3/04 IEEE.
7. Recep O. Ozdag, Peter A. Beerel (2004), 'A Channel Based Asynchronous Low Power High Performance Standard-Cell Based Sequential Decoder Implemented with QDI Templates', IEEE Proceedings of the 10th International Symposium on Asynchronous Circuits and Systems (ASYNC'04).
8. Riocereux.P.A, Brackenbury.E.M, Cumpstey.M, and Fruber.S.B (2001) 'A Low-Power Self-Timed Viterbi Decoder', in Proceedings of 7th International Symposium on Asynchronous Circuits and Systems.
9. [9] Viterbi.A (1967) 'Error bounds for convolutional codes and asymptotically optimum decoding algorithm', IEEE Transactions on Information theory, vol. It-13, no.2, pp. 60-269.
10. Wann-Shyang Ju, Ming-Der Shieh and Ming-Hwa Sheu (1997) 'A Low- Power VLSI Architecture for the Viterbi Decoder', IEEE proceedings.
11. Wuu and Sarma B. K. Vrudhula (1993) 'A design of a fast and area efficient multi- input Muller C-element', IEEE Transactions on VLSI Systems, pp.215-219.