

Chapters 6 and 11

Design Project: Designing a Viterbi Decoder (PART I)

1. Designing a Viterbi decoder

The Viterbi algorithm is commonly used in a wide range of communications and data storage applications. It is used for decoding convolutional codes, in baseband detection for wireless systems, and also for detection of recorded data in magnetic disk drives. The requirements for the Viterbi decoder or Viterbi detector, which is a processor that implements the Viterbi algorithm, depend on the applications where they are used. This results in very wide range of required data throughputs and power or area requirements. Viterbi detectors are used in cellular telephones with low data rates, of the order below 1Mb/s but with very low energy dissipation requirement. They are used for trellis code demodulation in telephone line modems, where the throughput is in the range of tens of kb/s, with restrictive limits in power dissipation and the area/cost of the chip. On the opposite end, very high speed Viterbi detectors are used in magnetic disk drive read channels, with throughputs over 600Mb/s. But at these high speeds, area and power are still limited.

In this semester's project we will design a critical part of a Viterbi decoder, under different design constraints.

1.1. The Viterbi Algorithm

The Viterbi algorithm is commonly expressed in terms of a trellis diagram, which is a time-indexed version of a state diagram. The simplest 2-state trellis is shown in Figure 1.

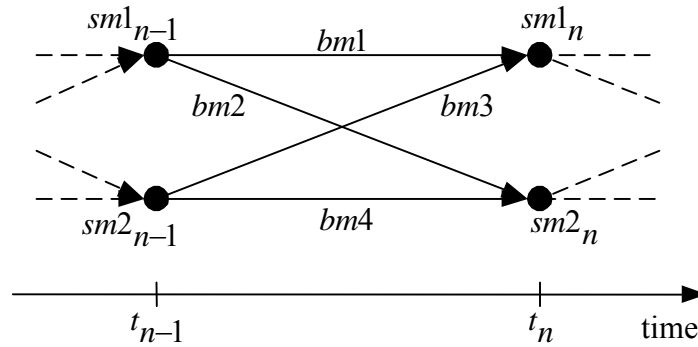


Figure 1: Two state trellis

The maximum likelihood detection of a digital stream with inter-symbol interference can be described as finding the most-probable path through a trellis of state transitions (branches). Each state corresponds to a possible pattern of recently received data bits and each branch of the trellis corresponds to the reception of the next (noisy) input.

The branch metrics represents the cost of traversing along a specific branch, as indicated in Figure 1. Under additive white Gaussian noise (AWGN conditions), it equals the squared difference between the received sample r , and the corresponding equalization target value t_k :

$$bm_k = (r - t_k)^2.$$

The state metrics, or path metrics, accumulate the minimum cost of 'arriving' into a specific state. The algorithm states are updated using an **add-compare-select** recursion. The branch metrics are added to the state metrics of the previous time instant. The smaller one of the two is selected to be the new state metric for each state, as illustrated in Figure 2

$$sm1_n = \min (sm1_{n-1} + bm1, sm2_{n-1} + bm3)$$

$$sm2_n = \min (sm1_{n-1} + bm2, sm2_{n-1} + bm4)$$

Figure 2: Add-compare-select recursion in the algorithm.

Finally, after all the input data is processed, the minimum state represents the *survivor* sequence. Tracing backwards we can then find the likely sequence of transmitted data.

An illustration of the Viterbi algorithm in operation using Java Applets can be found in: <http://www.alantro.com/viterbi/viterbi.htm>

1.2. Implementation of Viterbi Decoder

The implementation of the Viterbi decoder, a processor that implements the Viterbi algorithm, consists of three major blocks: the branch metrics calculation unit (BMU), the add-compare-select unit (ACS), and the survivor path decoding unit.

Branch metrics unit:

It performs the calculation of distances of sampled signals from targets, which are Euclidean in case of AWGN:

$$bm_{k,i} = (r_i - t_k)^2$$

(or Hamming in case of a binary symmetric channel).

The k new branch metrics are computed for each incoming sample r_i , at every clock cycle..

Add-Compare-Select:

A new value of the state metrics has to be computed at each time instant. In other words, the state metrics have to be updated every clock cycle. Because of this recursion, pipelining, a common approach to increase the throughput of the system, is not applicable. The Add-Compare-Select (ACS) unit hence is the module that consumes the most power and area.

In order to obtain the required precision, a resolution of 7 bits for the state metrics is essential, while 5 bits are needed for the branch metrics. Since the state metrics are always positive numbers and since only positive branch metrics are added to them, the accumulated metrics would grow indefinitely without normalization. In this project we have chosen to implement modulo normalization, which requires keeping an additional bit (8 instead of 7). The operation of the ACS unit is shown in Figure 3. The new branch metrics are added to previous state metrics to form the candidates for the new state metrics. The comparison can be done by using the subtraction of the two candidate state metrics, and the MSB of the difference points to a larger one of two.

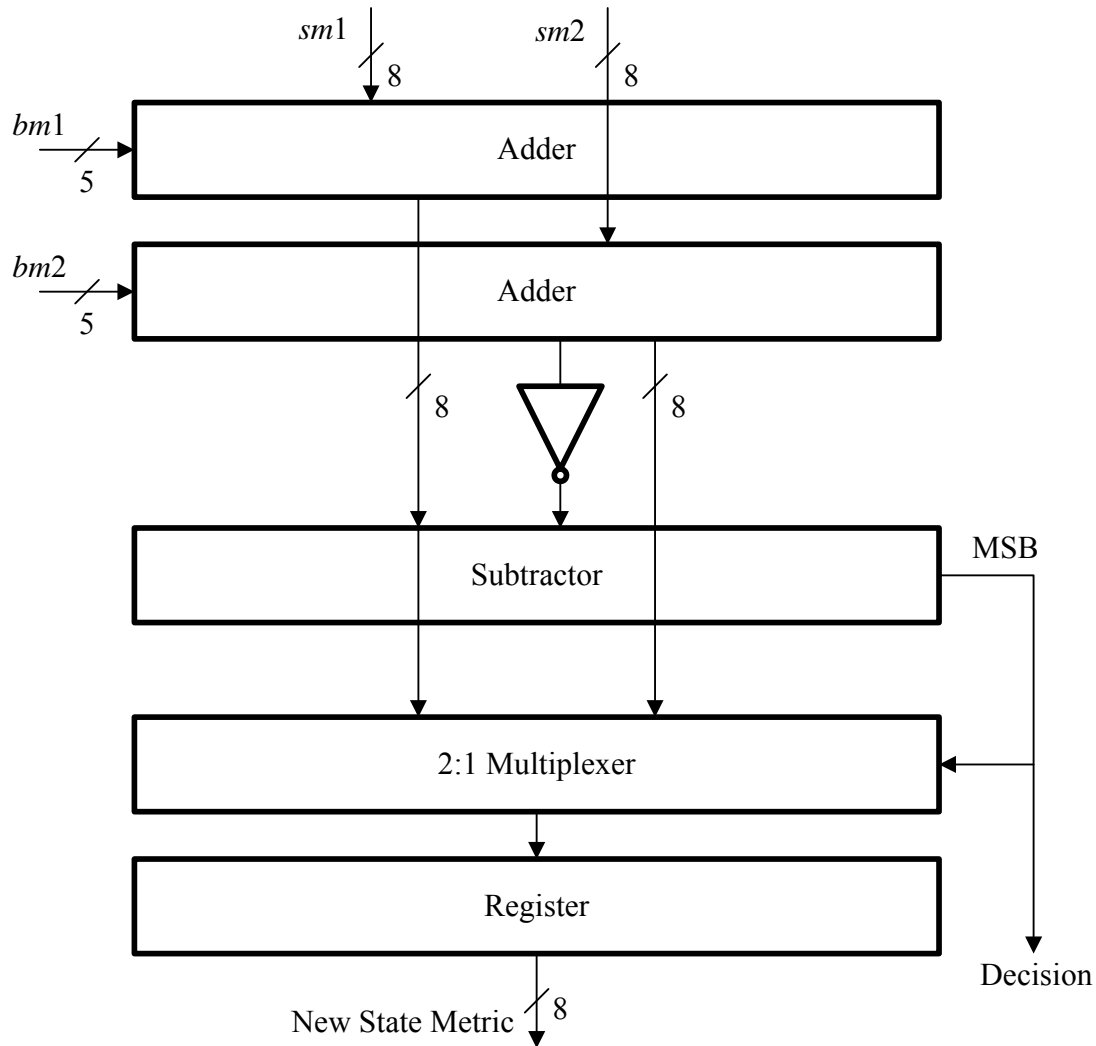


Figure 3: Block diagram of ACS unit.

Survivor sequence detection

In order to decode the input sequence, the survivor path, or shortest path through the trellis must be traced. The selected minimum metric path from the ACS output points the path from each state to its predecessor. In theory, decoding of the shortest path would require the processing of the entire input sequence. In practice the survivor paths merge after some number of iterations, as shown in bold lines in the 4-state example of Figure 4. From the point they merge together, the decoding is unique. The trellis depth at which all the survivor paths merge with high probability is referred as the survivor path length.

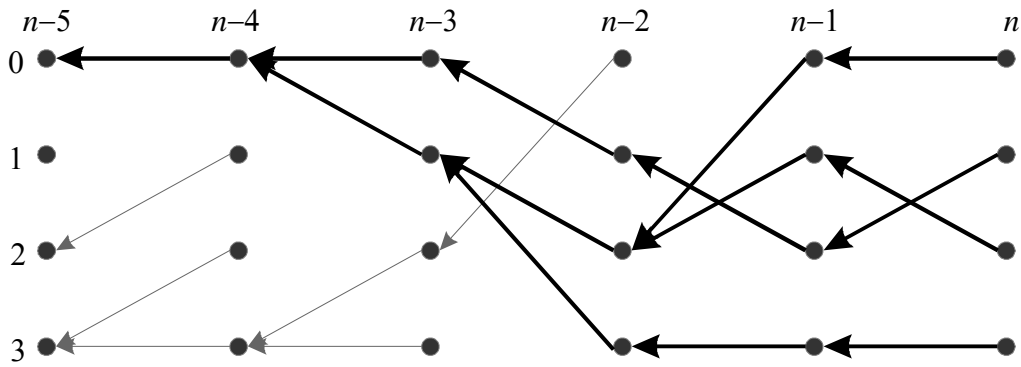


Figure 4: Survivor sequence detection.

2. Implementation and Constraints

The goal is to design an ACS unit to be used in the Viterbi decoder assuming one out of three scenarios

The project will be performed in THREE phases.

PHASE 1 GOALS:

The goal of the first phase is to perform the logic optimization, circuit style selection and first-order COMBINATIONAL circuit optimization to meet the stated design goals and constraints.

The fine-tuning of the design and the actual physical layout of the ACS will be performed in phase 2.

You should select ONE of the following THREE design scenarios:

- Low data throughput: Design a single ACS such that the average energy is minimized while still meeting the constraint that the worst-case delay is smaller than 50ns! No constraints are put on the area.**
- High data throughput: Maximize the single ACS operating speed. No constraints are put on area or power.**
- Low area decoder: Minimize the area of a single ACS, while meeting the constraint that the worst-case delay is smaller than 50ns! No constraints are put on energy.**

The project is to be done in pairs. You should sign up in teams of two students and choose design goal a) b) or c).

You are free to choose any logic family for the implementation of the project: complementary CMOS, pseudo-NMOS, pass-transistor logic, dynamic logic, etc.

TECHNOLOGY: The design is to be implemented in a 0.25 μm CMOS process with 4 metal layers. The SPICE technology is in the g25.mod file.

POWER SUPPLY: You are free to choose any supply voltage and logic swing up to 2.5V. Make sure that you use the appropriate model when you perform hand analysis.
PERFORMANCE METRIC: The propagation delays for static designs is defined as the time interval between the 50% transition point of the inputs and the 50% point of the worst-case output signal. Make sure you pick the worst-case condition and state **EXPLICITLY** in your report what that condition is. Note that for dynamic designs the propagation delay is defined in this case as the delay of the evaluate phase **ONLY** (at least in this phase of the process)!

AREA: The area is defined as **the smallest rectangular box** that can be drawn around the design.

NAMING CONVENTIONS: You should label the inputs and the outputs of the design as it is shown in Figure 3. The least significant bits of state metrics should be labeled as sm1[0] and sm2[0], and the most significant bits should be labeled as sm1[7] and sm2[7]. The least significant bits of branch metrics should be labeled as bm1[0] and bm2[0], and the most significant bits should be labeled as bm1[4] and bm2[4]. The newly computed state metric should be labeled as nsm[0]-nsm[7].

REGISTERS: In the first phase you don't need to design the registers. This will be a part of later phase of the project.

VOH, VOL, NOISE MARGINS: You are free to choose your logic swing. The noise margins should be at least 10% of the voltage swing. Test this by computing the VTC between one of the inputs and the output signals (with the other outputs set to the appropriate values) for a static design. For a dynamic circuit, apply an input signal with a 10% noise value added to the input and observe the outputs.

RISE AND FALL TIMES: All input signals and clocks have rise and fall times of 500ps. The rise and fall times of the output signals (10% to 90%) **should not exceed 1ns**.

LOAD CAPACITANCE: Each output bit of the ACS unit stage should have a **50 ff** load.

3. Layout

NO LAYOUT NEEDED IN THIS PHASE!

4. Simulation

You should demonstrate that your design is functionally correct (using IRSIM). Also, some first-order estimates of energy and performance should be provided.

5. Report

The quality of your report is as important as the quality of your design. One must sell the design by justifying the design decisions and providing all the vital information, while eliminating the unnecessary materials. **Organization, conciseness, and completeness are of paramount importance.** Use the templates provided on the web-page (in Framemaker, word, and pdf formats). Electronic submission of the reports is encouraged! If

filing electronically, e-mail your report as a postscript or pdf file. In case you do not have the means to create an electronic report, print out the template and deposit a paper copy.

Report Composition:

You should discuss your overall design philosophy and the important design decisions you made at the logic and circuit level. Discuss why your approach increases the operating speed or helps to reduce energy or area, while meeting the performance specs. Provide your current estimates of the results and describe how you got them. Include schematics and highlight the important elements.

Prove that your alleged results are TRUE by providing the crucial plots (don't forget to mention the input patterns you used to obtain those plots). **The total report should not contain more than three pages.** You are **not** allowed to add any other sheets, except for important plots. It should be based on the following outlay:

Page 1: Executive summary, overall design decisions, remarks and motivations

Page 2: Logic and transistor diagram - annotated with transistor sizes and worst-case timing path. Plot showing the functional operation of the cell. Comments.

Page 3: Timing and energy simulations - derive value of worst-case path and average energy. For the latter, a set of test patterns will be provided on the web page.

Also, you are required to send by e-mail the SPICE INPUT DECK you used to analyze the energy.

Remember, a good report is like a good layout: it should perform its function (convey information) **in the smallest possible area with the least delay and energy (to the reader) possible.**

Viterbi Decoder - Phase 2

1. Physical design of Viterbi ACS Unit

In the second phase of the project, you are to realize a physical design of the ACS unit of the Viterbi processor (that you designed in phase 1). The design should be laid out using Max.

Your layout must be free of design rule errors, and must include wells and sufficient contacts

to all these wells. Each input, output, and power supply wire should be brought to the edge of your cell with poly or any of the metal layers. Remember that you will be using this module in the third phase of the project. Some “thinking ahead” on how you will accomplish this is certainly advisable. For example, make sure that you plan carefully on how you will distribute the power lines through the design. Also, try to keep your design as regular as possible since a parameterizable and repetitive design is substantially more successful than a “spaghetti” circuit. **MODULAR DESIGN WILL EARN YOU EXTRA CREDIT IN THIS PROJECT!**

Use common sense in laying out your circuit and remember that long transistors must be built properly!

2. Updating of results

Most probably, mapping your design into a physical implementation will probably cause some important changes in the energy and delay numbers. Also, you have to ensure that your design is fully operational and correct. Hence, it is essential that you perform a full functional and performance analysis on the extracted circuit schematics. If you see major deviations from your results from phase 1, discuss why these are occurring. **Do not depart in a significant way from your original design or from your original goals.**

3. Report

You should discuss your overall layout strategy. Next, compare the results obtained from extraction with the ones you predicted earlier. Prove that your alleged results are TRUE by

providing the crucial plots (don't forget to mention the input patterns you used to obtain those plots). Mention any important changes you made with respect to phase 1.

The total report should not contain more than two pages. You are **not** allowed to add any other sheets, except for important plots. It should be based on the following outlay:

Page 1: Executive summary, overall design decisions, remarks and motivations

Page 2: Layout of the stage with indication of the terminals.

Also, you are required to send by e-mail the extracted SPICE INPUT DECK you used to analyze the energy.

Remember, the quality of the report is an important (major) part of the grade!

Viterbi Decoder - Phase 3

Chapter 7

1. Designing the register for the ACS Unit

In the third phase of the project, you are to design a register to be used with Viterbi ACS unit. You should pick a circuit topology of your choice that best meets the design goals that you have chosen in Phase 1 (speed, energy or area). You should also realize a physical design of the register using Max.

The clock signal is available with a rise/fall times of 500ps (10% to 90%). Each flip-flop should be loaded with 50fF load, and simulated under typical circuit conditions (same V_{DD} as the rest of the circuit, $T = 105$ degree C). You should report the Clk-Output delay of the flip-flop, setup and hold times, energy and area. Setup and hold times are defined as the intervals between the data and clock arrivals for which the Clk-Output delay deviates by 5% compared to its stationary delay. Assume input data signal rise and fall times of

500ps. When doing the physical design, take into consideration that you should be able to place it together with the combinational portion of the ACS unit.

2. Simulation of the complete ACS unit

In order to test your circuit under the fair conditions in this final phase, you should place the 8-bit register consisting of 8 flip-flops that you designed at the output of the combinational part of the ACS unit from phase 2. Connect the register outputs to the *sm1* inputs of the ACS, as shown in Figure 1. You should simulate the extracted circuit to determine the maximum operating frequency of your ACS unit. **Do not modify the existing ACS layout unless it is necessary!**

Initialization of the flip-flop output voltage levels may help convergence of your simulations. Note that assumed loadings may change, according to your design, and may cause some important changes in the energy and delay numbers. Also, you have to ensure that your design is fully operational and correct. Hence, it is essential that you perform a full functional and performance analysis on the extracted circuit schematics. If you see major deviations from your results from phase 2, discuss why these are occurring. **Do not depart in a significant way from your original design or from your original goals.**

TIP: In order to simplify the extraction process, you can extract only the eight-bit register with appropriate wire loading, and use the previously extracted ACS unit for simulations. Create an instance of the previously completed ACS in your top level cell. Do NOT flatten. Realize the physical layout of the registers and the feedback wiring. When you are ready for simulation, remove the instance of ACS unit. Extract the remaining registers and wiring. Finally, connect this to a complete ACS extraction as a module.

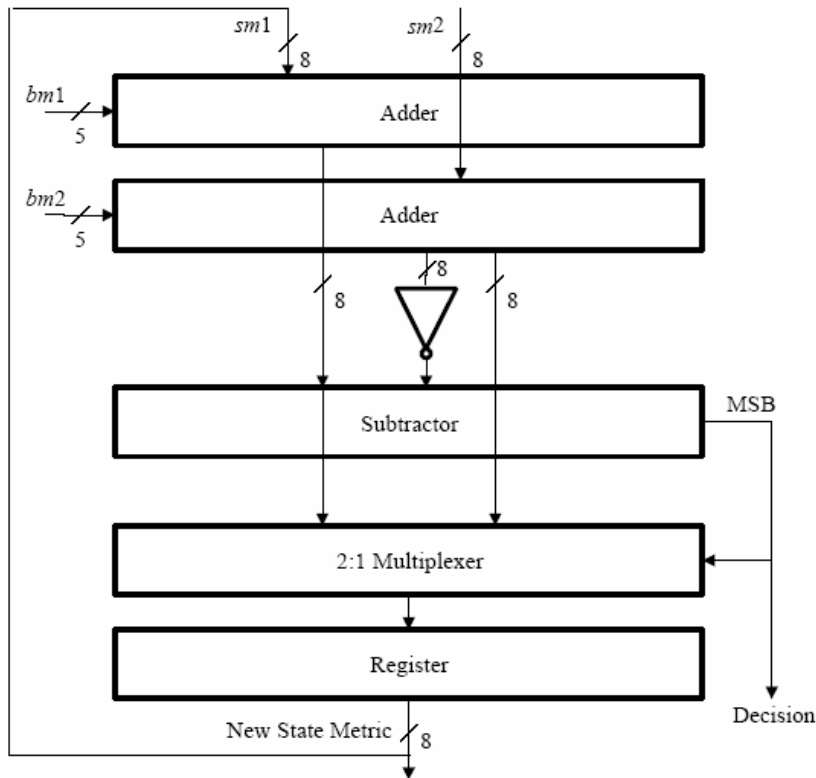


Fig.1: ACS unit with register

3. Report

You should discuss your overall design and layout strategy. Discuss your choice of the flip-flop. Next, compare the results obtained from extraction of the complete ACS unit design with the ones you predicted earlier by separate designs of combinatorial and sequential blocks. Prove that your alleged results are TRUE by providing the crucial plots (don't forget to mention the input patterns you used to obtain those plots). Mention any important changes you made with respect to phase 1 and phase 2.

The total report should not contain more than three pages. You are **not** allowed to add any other sheets, except for important plots. It should be based on the following outlay:

Page 1: Executive summary, overall design decisions, remarks and motivations

Page 2: Layout of the single flip-flop and the whole ACS unit with the register with indication of the terminals.

Page 3: Timing and energy simulations. Flip-flop Clk-Q delay, setup and hold times, energy. Derive the value of worst-case path delay for the complete design. Include the plot that illustrates the following conditions: initialize the values of $sm1$ to equal 11111110, and fix the values of $sm2$ to all ones and $bm2$ to all zeroes. Bring the value of 00001 to input $bm1$ and provide the graph with the values of $sm1[0] - sm1[7]$ in three consecutive clock cycles (illustrating the worst case delay).

Provide the comparison of the worst case delays through add, compare, select and register blocks with the minimum cycle time. Prove that the hold time requirement is met. You do not have to perform energy simulations of the whole design.

Also, you are required to send by e-mail the extracted SPICE INPUT DECK you used to analyze the minimum cycle time.

GOOD LUCK!