

應用於無線通訊之低功耗基頻處理器(1/3)

A Low-Power Baseband Processor for Wireless Communication System (1/3)

計畫編號: NSC97-2220-E-009-166

執行期間: 97 年 8 月 1 日 至 98 年 7 月 31 日

主持人: 李鎮宜 交通大學電子工程系教授

一、中文摘要

基頻訊號處理在無線通訊系統上扮演關鍵性的角色，不僅可有效提升傳輸的效能，更能提供多模式和多標準的系統實現方案。然而要達成低成本和低功耗設計方法，不僅對於個別模組的演算法需深入瞭解外，也必須融入系統層級的行為，方能提供一具有技術競爭力的解決方案。因此在這一年的研究計畫，我們針對 OFDM 主流無線通訊系統所需求的關鍵模組 Viterbi decoder 跟 BCH decoder 進行相關議題的研究，探討低功耗 Viterbi decoder 以及低複雜度的軟性 BCH decoder 的設計方式，並研究在不同設計規範下達成多模和多標準的作業模式，之後將會把此關鍵模組設計整合，並配合基頻訊號的同步模組電路，完成一符合多模、多標準的低功耗基頻處理器。

關鍵字

基頻處理器；多模式；多標準；低成本；低功耗

英文摘要

Signal processing in baseband processor designs plays a key role in wireless communication system designs—in not only improving overall system transmission performance, but also providing the capability of multi-mode and multi-standard for cost-effective system realization. To reach better performance indices in terms of low-cost and low-power, it is necessary to investigate system design methodologies, covering in-depth exploration of algorithms of key modules and exploitation of unique features/behaviors of a complete system. As a result, a more competitive solution can be delivered. In this year (2008/8~2009/7), we have concentrated on the key modules (Viterbi decoder and BCH decoder) of the main-stream OFDM wireless communication systems. The first issue is the low-cost solution for Viterbi decoder. The second issue is the low-cost solution for soft BCH decoder. In the end, these design techniques and key modules will be integrated on a design platform, together with synchronization modules, to come up with a multi-mode, multi-standard, and low-power baseband processor.

Keywords

Baseband Processor, Multi-mode, Multi-Standard, Low-Cost, Low-Power

二、計畫的緣由與目的

A. A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation

Length

The Viterbi decoder implementing the Viterbi algorithm [A-1] for decoding convolutional codes is composed of three main blocks: the branch metric (BM) unit, the ACS unit, and the survivor memory. The BM unit generates branch metrics from the input data. The ACS unit recursively accumulates branch metrics as path metrics (PM) and makes decisions to select the most likely state transition sequences, or the survivors. Survivor memory stores the survivors for retrieving the data sequence.

There are two well-known survivor memory management approaches: the register-exchange (RE) and the traceback (TB) [A-2]. The register-exchange is conceptually the simplest technique that eliminates repeatedly memory access operations. Therefore, this approach has shorter latency and is suitable for high speed decoder implementations. However, due to the data movement among registers, the approach is considered to be power inefficient.

Figure 1 shows the conventional 2^v -state Viterbi decoder with the register-exchange architecture [A-3]. The decisions from ACS units will be shifted within the survivor memory from left to right. Applying the scarce state transition (SST) technique and the variable truncation length, we illustrate the proposed low-power Viterbi decoder for the MB-OFDM UWB system [A-4] in Figure 2. The SST unit is integrated to reduce state transition activities, leading to less dynamic power consumption [A-8]. Furthermore, the path merging detector monitors the merged point for all survivors and adjusts the truncation length to avoid unnecessary data movement in registers. Many redundant operations in the survivor memory can be reduced to save power dissipation. Additionally, considering the high throughput requirement, the radix-4 ACS structure is exploited because of a better compromise between cost and throughput [A-5].

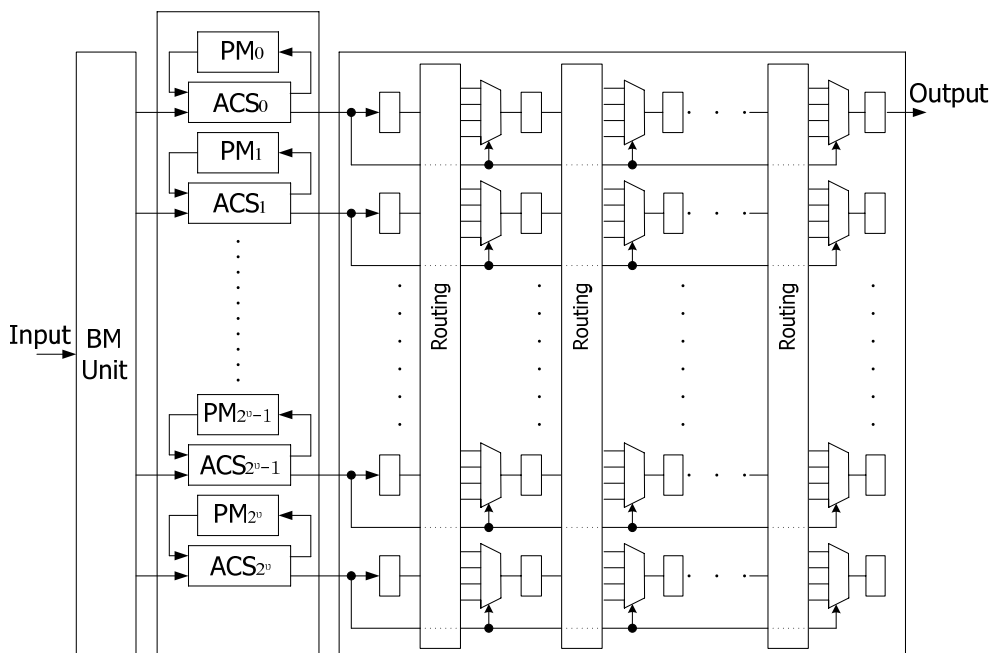


FIGURE 1: THE CONVENTIONAL REGISTER-EXCHANGE ARCHITECTURE.

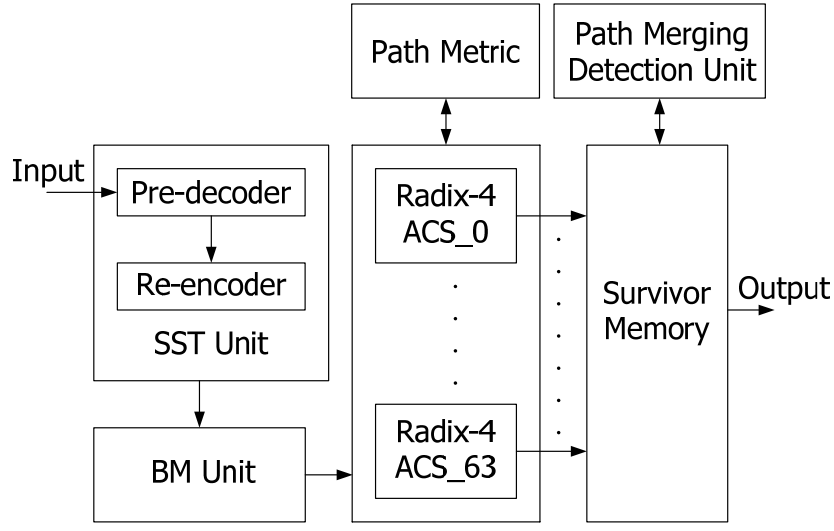


FIGURE 2: THE PROPOSED VITERBI DECODER ARCHITECTURE.

For the Viterbi decoder, the scarce state transition (SST) algorithm is a low power technique to reduce the state transition activity significantly under high SNR conditions [A6-A8]. In the conventional Viterbi decoding model (see Figure 3(a)), $\mathbf{u}(D)$ denotes the information sequence, $\mathbf{C}(D)=\mathbf{u}(D)G(D)$ is the codeword sequence deriving from the generator polynomial $G(D)$. From the received sequence $\mathbf{r}(D)$, the Viterbi decoder estimates the decoded information $\mathbf{o}(D)$.

The SST Viterbi decoding architecture in Figure 3(b) includes two additional blocks: pre-decoder and re-encoder. Assume

$$\mathbf{r}(D) = \mathbf{u}(D) \cdot G(D) + \mathbf{e}(D) = \mathbf{C}(D) + \mathbf{e}(D) \quad (1)$$

and $\mathbf{e}(D)$ is the error sequence from a noisy channel, the pre-decoder directly decode the information sequence from $\mathbf{r}(D)$:

$$\mathbf{i}(D) = \mathbf{r}(D) \cdot G^{-1}(D) = \hat{\mathbf{u}}(D) \quad (2)$$

The re-encoder then encodes $\mathbf{i}(D)$ to a new codeword $\mathbf{z}(D)$.

$$\mathbf{z}(D) = \mathbf{i}(D) \cdot G(D) = \hat{\mathbf{C}}(D) \quad (3)$$

The Viterbi decoder performs maximum likelihood decoding on $\mathbf{y}(D)$, which is defined as follows:

$$\mathbf{y}(D) = \mathbf{r}(D) + \mathbf{z}(D) = \mathbf{C}(D) + \mathbf{e}(D) + \hat{\mathbf{C}}(D) \quad (4)$$

In high SNR conditions, $\mathbf{e}(D)$ is nearly zero, and the decoded information sequence becomes

$$\mathbf{o}(D) = \mathbf{i}(D) + \mathbf{n}(D) = \hat{\mathbf{u}}(D) + \mathbf{n}(D) \quad (5)$$

If the channel condition is good enough, the decoder estimates an approximately zero sequence; as a result, the dynamic power is reduced as the channel becomes better.

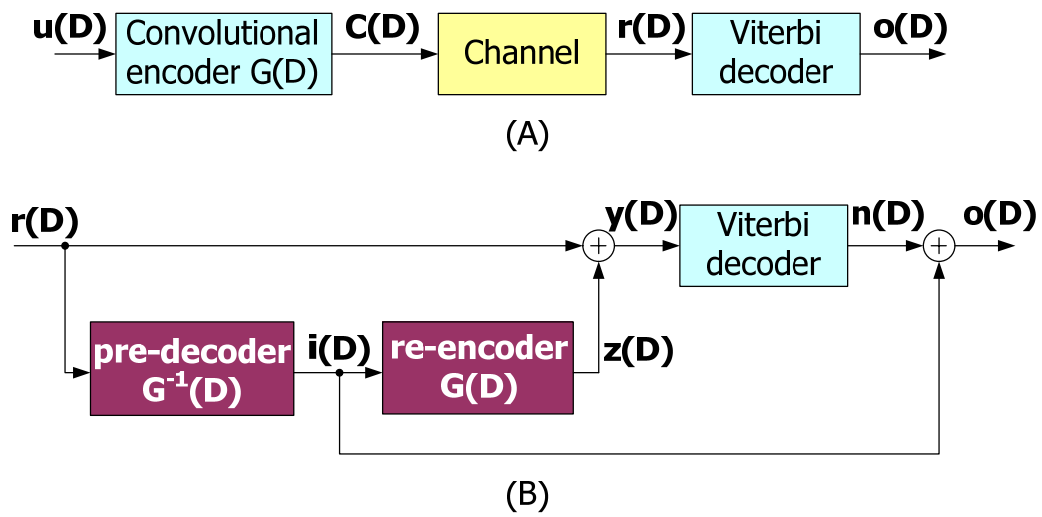


FIGURE 3: THE (A) CONVENTIONAL AND (B) SST DECODING MODEL.

B. Soft BCH Decoders Using Error Magnitudes for Digital Video Broadcasting

The Bose-Chaudhuri-Hocquenghen (BCH) [B-1] codes are popular in storage and communication systems recently. From DMB-T [B-2] and DVB-S2 [B-3] applications shown in Figure 4, the BCH codes with long block length are specified to suppress the error floor due to iterative LDPC decoding. Since BCH codes perform as outer codes in those communication systems, the soft information from the inner decoder can be employed to further improve the error-correcting performance.

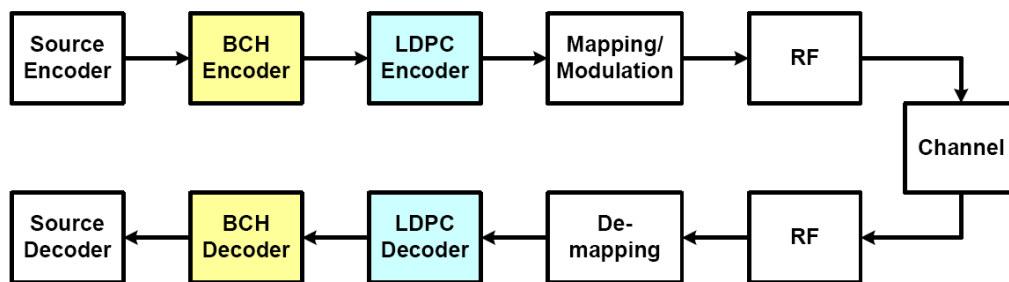


FIGURE 4: BLOCK DIAGRAM OF DMB-T AND DVB-S2 SYSTEMS

Soft decision decoding of BCH codes has aroused many research interests. Forney developed the generalized-minimum-distance (GMD) [B-4], which uses algebraic algorithms to generate a list of candidate codewords and chooses a most likely codeword from the list. Other algorithms with the same concept of candidate list, such as Chase [B-5] and SEW [B-6], are also widely used in many applications. This report illustrates a soft BCH decoding method using error

magnitudes [B-7] to deal with the least reliable bits. For example, Figure 5 shows the results of a concatenated code with 16-state BCJR [B-8] and BCH (255,239) over GF(2⁸). Based on the soft information from previous decoder, the performance gain of the BCH decoder is about 0.73 db at BER = 10⁻⁶ when 2t+1 candidate bits within a codeword are chosen to correct errors.

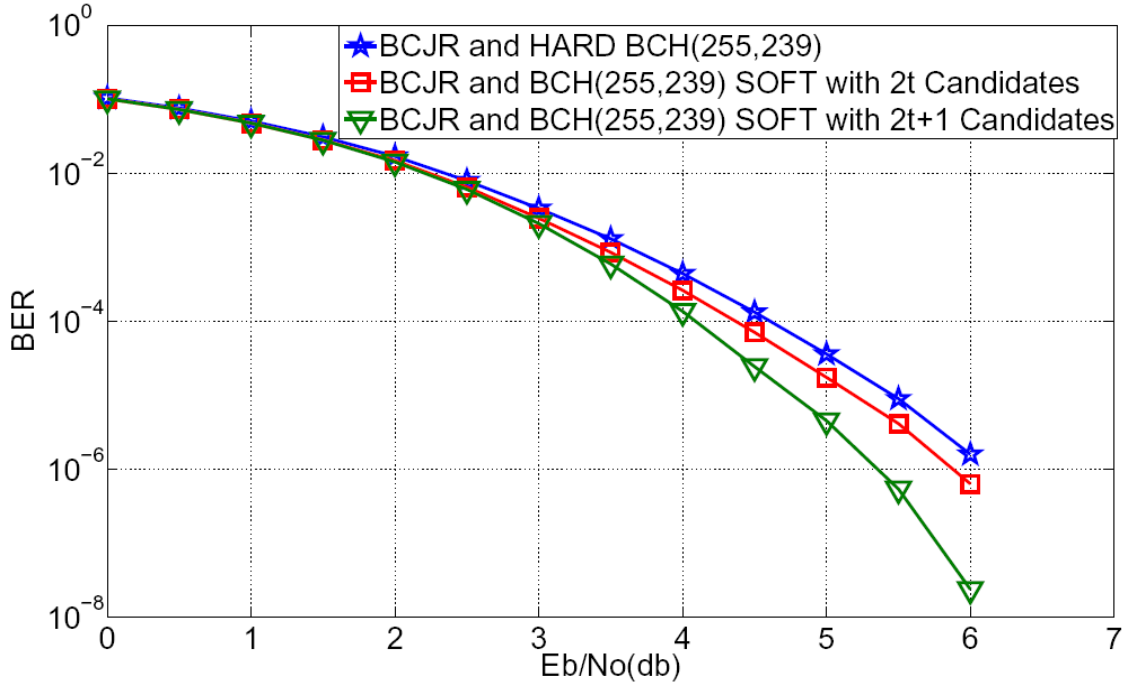


FIGURE 5: SIMULATION RESULTS FOR BCH (255,239) CONCATENATING WITH A 16-STATE BCJR UNDER BPSK MODULATION AND AWGN CHANNEL.

三、 研究方法及成果

A. A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length

1. Power Reduction with Variable Truncation Length

As indicated in the Viterbi algorithm, the decoder output is the codeword that minimizes the conditional probability of the received sequence. Therefore, the entire received sequence should be stored and analyzed before any decoding output. Nevertheless, the received sequence length may be large, and the survivor path should be truncated to reduce storage requirement and decoding latency. If the truncation length T is large enough, about five times constraint lengths, the performance can achieve that of maximum-likelihood decoding.

Figure 6 illustrates the survivor paths stored in a survivor memory. All the 2^v survivor paths will merge with a high probability for a 2^v -state Viterbi decoder. Consequently, it is more efficient to store the merged path rather than the 2^v paths after the merged stage. The truncation lengths depend strongly on the channel conditions as listed in Table 1. Based on the variable

truncation length property [A-9], we design a path merging detection unit to reduce the power consumption in the survivor memory.

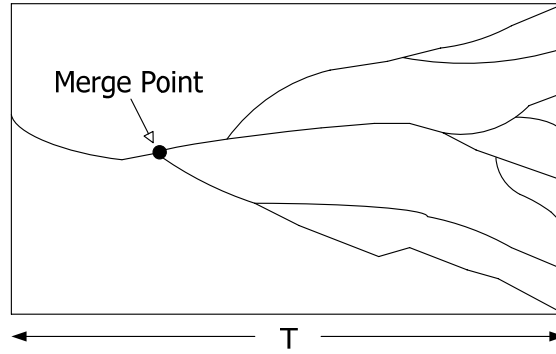


FIGURE 6: SURVIVOR PATHS STORED IN THE SURVIVOR MEMORY.

TABLE 1: AVERAGE REQUIRED TRUNCATION LENGTH FOR PATH MERGING IN DIFFERENT CHANNEL CONDITION.

E_b/N_0	1.0	2.0	3.0	4.0	5.0
Truncation length	33.78	26.86	23.34	21.54	20.54

2. VARIABLE TRUNCATION LENGTH

The Viterbi decoder for the MB-OFDM UWB system has 64 states. Figure 7 illustrates the survivor memory on the radix-4 trellis. D_0 to D_{63} are the decisions provided by the ACS units for selecting survivor paths. Base on path merging property, the 64 states tend to be equivalent from the left stages to the right stages, which are more reliable.

The path merging detection unit will find the merge point, or stage in the trellis. Obviously, if contents of all the 64 survivors are equivalent at the same stage, the 64 survivor paths have merged. However, it is complex to check all 64 states concurrently. To reduce the hardware complexity, our proposal detects path merging by dividing 64 states into 16 groups that are verified separately. The simulation results show that this scheme has no performance loss. We assume the 64 survivor paths have merged and the value in state 0 is already reliable if every group (the circles in Figure 7) contains equivalent values at the same stage. After detecting the merged point, we apply clock gating to the registers in the shadow region and directly shift out the value. The state 0 path is considered as the correct one, and the others are dropped.

Figure 8 illustrates the survivor memory architecture with variable truncation length. The registers of each stage are connected to the path merging detection unit that decides the merge point and generates clock gating signals of each stage. Based on the scheme, we can adjust truncation length dynamically, depending on the channel. In high SNR environments, a shorter truncation length is required and the clock gating can be applied to more registers, resulting in a power efficient survivor memory.

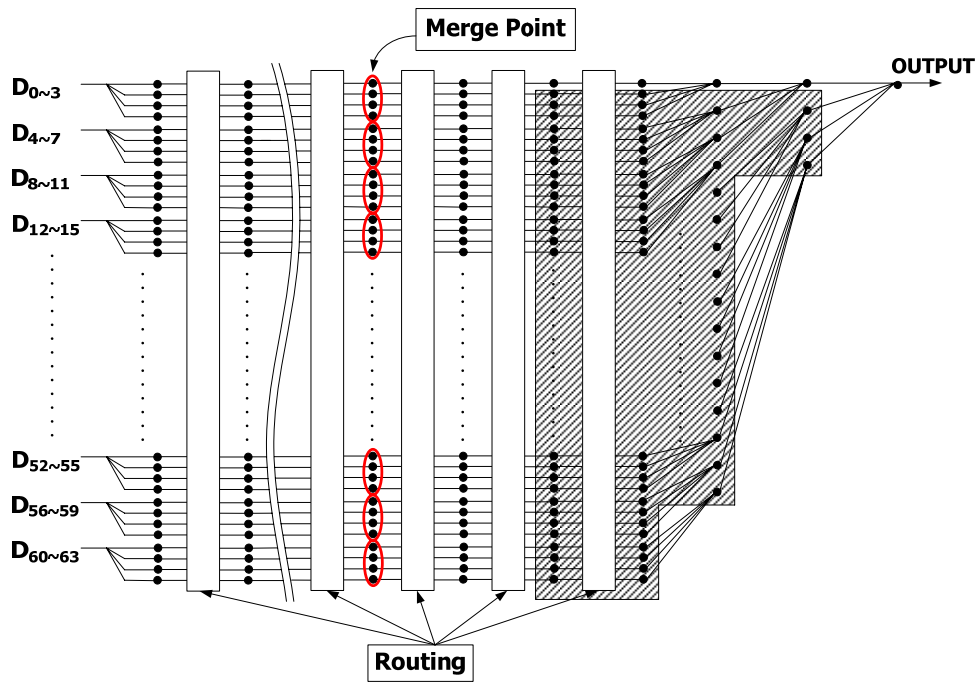


FIGURE 7: PATH MERGING DETECTION SCHEME.

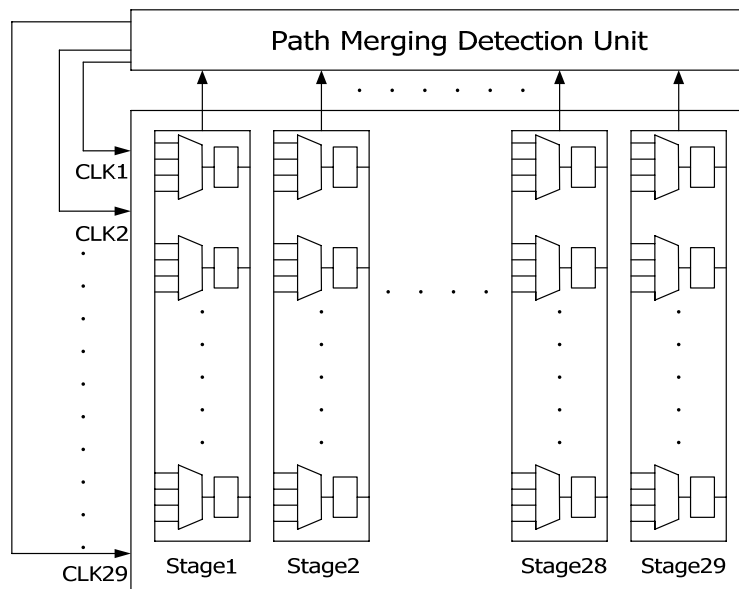


FIGURE 8: SURVIVOR MEMORY ARCHITECTURE WITH VARIABLE TRUNCATION LENGTH.

4. IMPLEMENTATION RESULTS

3. DESIGN PARAMETERS AND PERFORMANCE SIMULATION

The Viterbi decoder, based on the register-exchange approach, combines the SST and the path merging detection schemes to reduce power dissipation. The design parameters of the proposed Viterbi decoder are listed in Table 2.

Figure 9 shows the performance simulation result in the BPSK modulation. Notice that the performance of the conventional scheme, the SST scheme, and the proposed scheme are approximately the same. Compared with the floating point case, the performance degradation of 8-level soft-decision is less than 0.5dB. The proposed variable truncation length scheme still preserves the error performance.

TABLE 2: DESIGN PARAMETERS OF THE PROPOSED VITERBI DECODER.

Technology	1.2V 0.13- μm 1P8M CMOS
State Number	64
Code Rate	1/3
Soft-Decision	8-levels
BM Width	6 bits
PM Width	9 bits
Max. Truncation length	64
ACS structure	radix-4

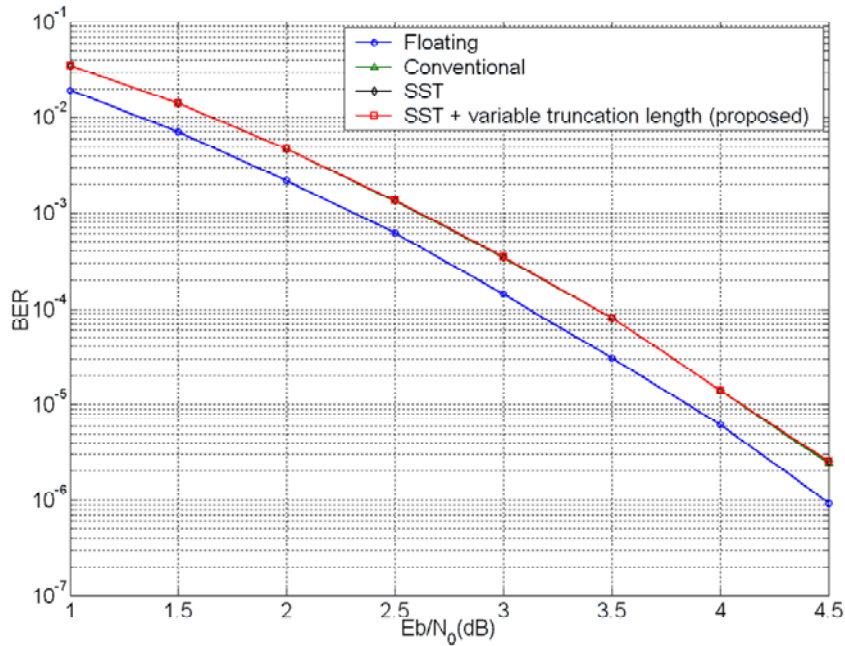


FIGURE 9: SIMULATION RESULTS IN AWGN CHANNEL, BPSK, 8-LEVEL SOFT-DECISION AND CODE RATE = 1/3.

4. POWER SIMULATION

We analyze the power dissipation of three implementations: the conventional register-exchange approach, the SST scheme without and with the variable truncation length scheme. Table 3 lists the gate counts of these implementations. In different channel environments, we compare the power consumption of the three structures. Figure 10(a) and Figure 10(b) respectively reveal the post-layout power estimation of the whole Viterbi decoder and the survivor memory. The operating frequency is 250MHz and the corresponding data rate is 500Mbps due to the radix-4 ACS structure.

For the conventional design, the channel conditions are ineffective in the power dissipation. In the SST only implementation, the decoder power dissipation is reduced in high SNR environments; however, the power reduction is not obvious due to the complex signal wire routing. In the proposed design combining the SST and the variable truncation length, the decoder power has a significantly reduction as shown in the figures. Figure 10(b) shows survivor

memory power only to highlight the effect of the dynamic truncation length. As the channel condition is good enough, the variable truncation length scheme lowers more than 60% survivor memory power.

Figure 11 shows the power profiling of the conventional register-exchange structure and the proposed decoder as E_b/N_0 is 5.0 dB. The corresponding bit error rate in this channel condition is $2.56e-6$. From Figure 11(a), the survivor memory is a power intensive block in conventional decoder designs. With SST and variable truncation length schemes, the ratio of survivor memory power is reduced significantly (see Figure 11(b)). Furthermore, the SST unit and the path merging detection unit consume less than 2% of the decoder power.

TABLE 3: THE GATE COUNTS OF DIFFERENT IMPLEMENTATIONS.

Implementation	Gate counts
Conventional RE approach	108.5k
SST scheme	109.0k
Proposed	116.9k

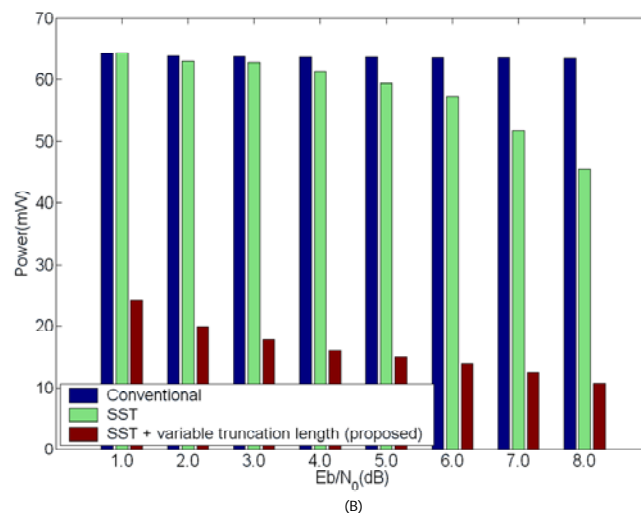
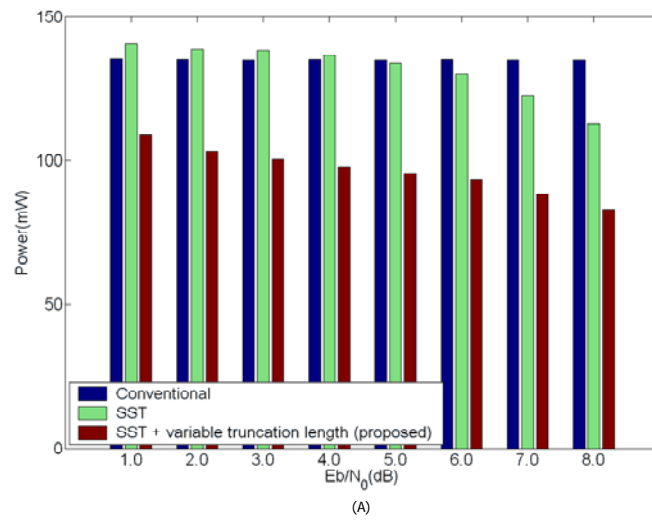


FIGURE 10: COMPARISON OF (A) DECODER POWER (B) SURVIVOR MEMORY POWER AT 500MBPS.

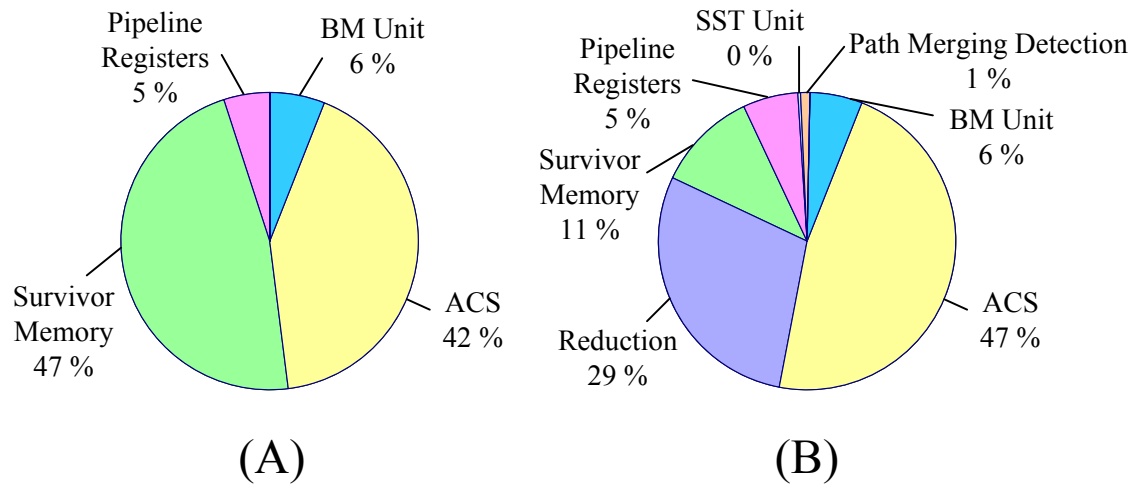


FIGURE 11: THE POWER PROFILING OF (A) CONVENTIONAL RE STRUCTURE AND (B) PROPOSED STRUCTURE AS EB/N_0 IS 5.0 DB.

B. Soft BCH Decoders Using Error Magnitudes for Digital Video Broadcasting

1. Soft Decision BCH Decoding

Conventional BCH decoding contains syndromes calculation, key equation solver, and *Chien search* [B-1]. In general, the complexity of a soft BCH decoder is much higher than a hard BCH decoder for decoding an entire codeword. Nevertheless, soft BCH decoders with lower complexity can be revealed by focusing on the least reliable bits instead of the whole codeword.

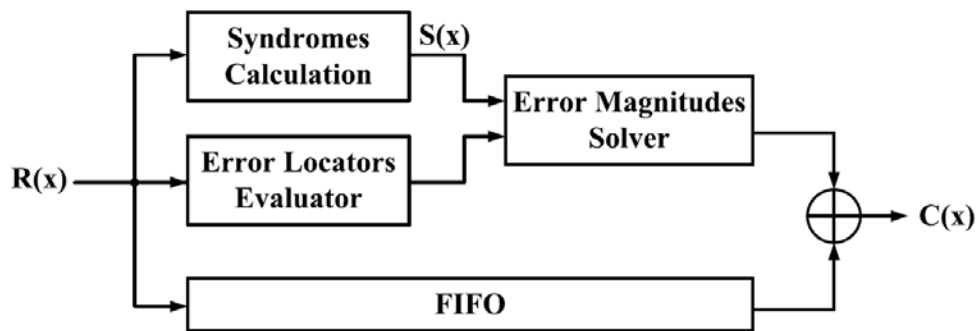


FIGURE 12: SOFT DECISION BCH DECODING BLOCK DIAGRAM

As shown in Figure 12, the soft BCH decoding using error magnitudes [B-7] includes three major steps: *syndromes calculation*, *error locators evaluator*, and *error magnitudes solver*. From the received polynomial $R(x)$, the syndromes polynomial $S(x) = S_1 + S_2x^1 + \dots + S_{2t}x^{2t-1}$ are expressed as

$$S_j = R(\alpha^j) = \sum_{i=1}^v (\alpha^j)^{l_i} = \sum_{i=1}^v (\beta_{l_i})^j \quad (1)$$

For $j = 1, 2, \dots, 2t$, where α is the primitive element over $GF(2^m)$. Notice that l_i is the i -th actual error location and $\beta_{l_i} = \alpha^{l_i}$ indicates the corresponding error locator. With soft inputs, error locators evaluator can choose $2t$ least reliable inputs and evaluates their corresponding error locator values to form a β -vector, $[\beta_{c_1}, \beta_{c_2}, \dots, \beta_{c_{2t}}]$. Also, the error location set, $\{L_{c_1}, L_{c_2}, \dots, L_{c_{2t}}\}$, can be calculated with β -vector because the β value of the L_{c_i} -th location is $\beta_{c_i} = \alpha^{L_{c_i}}$. The relation between β_{c_i} and the syndromes can be formulated as

$$\begin{bmatrix} \beta_{c_1} & \beta_{c_2} & \cdots & \beta_{c_{2t}} \\ \beta_{c_1}^2 & \beta_{c_2}^2 & \cdots & \beta_{c_{2t}}^2 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{c_1}^{2t} & \beta_{c_2}^{2t} & \cdots & \beta_{c_{2t}}^{2t} \end{bmatrix} \begin{bmatrix} \gamma_{c_1} \\ \gamma_{c_2} \\ \vdots \\ \gamma_{c_{2t}} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{2t} \end{bmatrix} \quad (2)$$

where γ_{c_i} is the error magnitude corresponding to β_{c_i} for $i = 1, 2, \dots, 2t$. The left $2t \times 2t$ matrix in (2) is defined as β -matrix. From (1) and (2), it is evident that if all the errors are in the location set, the exact γ_{c_i} value can be determined; otherwise, this decoding approach fails to correct errors. The error magnitudes solver shown in Figure 12 is used to solve (2) to get γ_{c_i} . For those γ_{c_i} equal to 1, the corresponding L_{c_i} are the exact error locations. The codeword polynomial $C(x)$ can be obtained by inverting the L_{c_i} -th values in the received polynomial $R(x)$.

2. Proposed Algorithm and Architecture

I. Error Locators Evaluator

As shown in Figure 13, error locators evaluator architecture includes *the reliability part*, *the error locator part* and *the error location part*. The upper part is the reliability part which stores the reliabilities of $2t$ least reliable candidates $R_{c_1}, R_{c_2}, \dots, R_{c_{2t}}$. The medium part is the error locator part to construct the β -vector. Because the β value of the i -th location is α^i , the β value of $(i+1)$ -th locations is α times the β values of i -th location. The β value can be computed by multiplying α^{-1} with register REG if the input is serial in from the highest degree coefficient of $R(x)$. Thus, the error locator part can use a constant multiplier to calculate the error locator of each input. Notice that register REG initially contains the β value of the first input. The bottom part is the error location part. The decoding method focuses on the least reliable bits instead of the whole codeword, so the error location part uses a counter to compute the error location L_{c_i} corresponding to each R_{c_i} for serial input. Hence, the Chien search procedure is no longer

required and a lot of redundant decoding latencies can be eliminated.

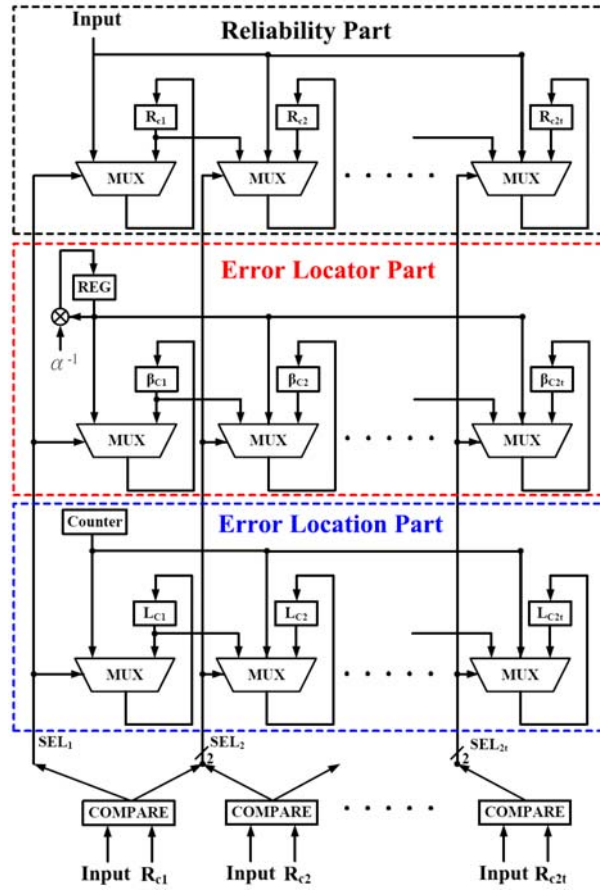


FIGURE 13: ERROR LOCATORS EVALUATOR ARCHITECTURE FOR SERIAL INPUT

Error locators evaluator classifies the soft inputs to choose $2t$ least reliable inputs as the candidate reliabilities $R_{c1}, R_{c2}, \dots, R_{c2t}$. Their corresponding error locators β_{ci} and error locations L_{ci} are also calculated and stored in registers. Error locators evaluator compares the soft inputs with R_{ci} , and then generate the select signals SEL_i to control the multiplexers. In the i -th stage, if the input is smaller than $R_{c_{i-1}}$, the i -th stage is updated with $(i-1)$ -th stage value. If the input is greater than $R_{c_{i-1}}$ and smaller than R_{ci} , the i -th stage is updated with the input value. Otherwise, the i -th stage holds its current value.

II. Error Magnitudes Solver (EMS)

To obtain the valid γ_{ci} value in (2), the *Gauss Elimination* method is the most intuitive way but the complexity is $O(n^3)$. Two alternative algorithms for improving decoding efficiency under different error correcting capabilities t are revealed. One uses the characteristic that the valid error magnitude in BCH codes is either 0 or 1, and the other employs the quick Vandermonde matrix solution.

i. Heuristic Error Magnitudes Solver(H-EMS):

In BCH codes, the valid error magnitude in (2) is either 0 or 1, so the problem can be formulated into checking all combinations of γ_{ci} over GF(2) instead of calculating real error magnitudes. A 2t-bit counter is used to do a heuristic search for all binary combinations. Since $S_1^2 = S_2$, $S_2^2 = S_4$, . . . , $S_t^2 = S_{2t}$ in BCH codes, the even part of syndromes check can be eliminated to simplify (2) as :

$$\begin{bmatrix} \beta_{c_1} & \beta_{c_2} & \cdots & \beta_{c_{2t}} \\ \beta_{c_1}^3 & \beta_{c_2}^3 & \cdots & \beta_{c_{2t}}^3 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{c_1}^{2t-1} & \beta_{c_2}^{2t-1} & \cdots & \beta_{c_{2t}}^{2t-1} \end{bmatrix} \begin{bmatrix} \gamma_{c_1} \\ \gamma_{c_2} \\ \vdots \\ \gamma_{c_{2t}} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_3 \\ \vdots \\ S_{2t-1} \end{bmatrix} \quad (3)$$

TABLE 4: THE PROPOSED HEURISTIC EMS ALGORITHM

<p><i>Input</i> : a 2t – bit CNT : {CNT₁, CNT₂, . . . , CNT_{2t}} = 0 syndromes vector [S₁, S₃, . . . , S_{2t-1}], and β_{c_i} matrix for (k = 1; k < 2^{2t}; k++) for (j = 1; j < 2t; j=j+2) $\delta_j = \sum_{i=1}^{2t} CNT_i \times \beta_{c_i}^j$ if (all $\delta_j == S_j$) Decoding Successes !!! else CNT = CNT + 1</p>

Table 4 illustrates details of the proposed H-EMS algorithm. The i-th bit of CNT, CNT_i, performs as the i-th error magnitude, γ_{ci} . Thus, by iteratively flipping each CNT_i value, a heuristic search for all binary combinations can be completed. At each iteration, the solver can verify where the equation (3) stands or not. As shown in Figure 14, H-EMS uses 2t(t-1) multipliers to construct the β -matrix. Each $\beta_{c_i}^j$ value will be calculated with CNT_i, and the solver checks the results equal to the syndromes or not.

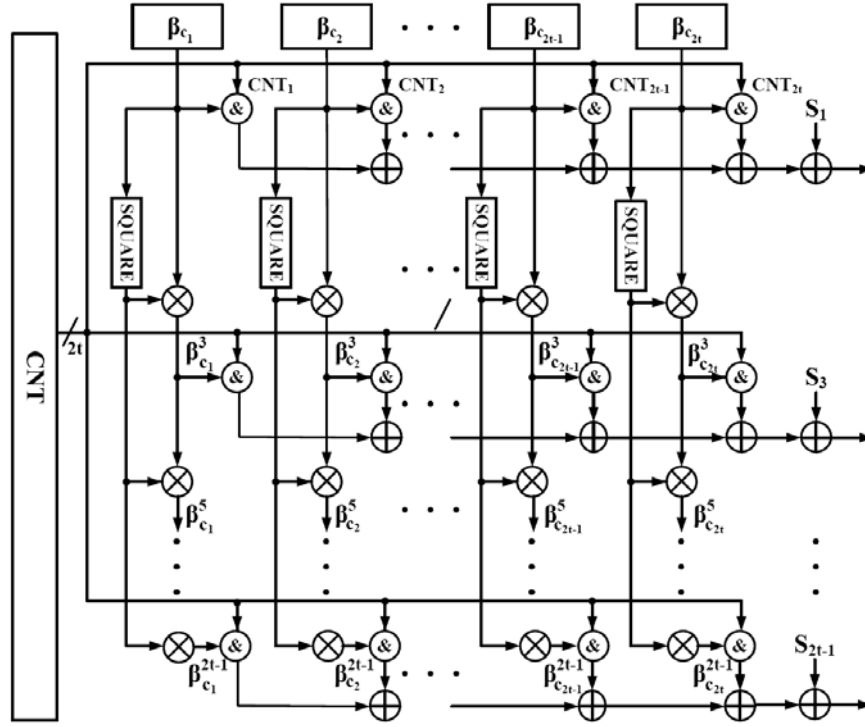


FIGURE 14: HEURISTIC ERROR MAGNITUDES SOLVER ARCHITECTURE

ii. **Borck-Pereyra Error Magnitudes Solver(BP-EMS)**

TABLE 5: BORCK-PEREYRA ALGORITHM

<p><i>Input</i> : syndromes vector $[S_1, S_2, \dots, S_{2t}]$ β_{c_i} vector $[\beta_{c_1}, \beta_{c_2}, \dots, \beta_{c_{2t}}]$ for ($k = 1; k < 2t, k++$) for ($i = 2t; i > k, i--$) $S_i = S_i - \beta_{c_k} S_{i-1}$ for ($k = 2t - 1; k > 0, k--$) for ($i = k + 1; i \leq 2t, i++$) $S_i = S_i / (\beta_{c_i} - \beta_{c_{i-k}})$ for ($i = k; i < 2t, i++$) $S_i = S_i - S_{i+1}$ for ($k = 1; k \leq 2t, k++$) $S_i = S_k / \beta_{c_k}$ <i>Output</i> : S_i is error magnitude</p>
--

Since the β_{c_i} matrix is a Vandermonde matrix, Borck-Pereyra algorithm [B-9] [B-10] shown in Table 5 can calculate the error magnitudes efficiently for large matrix. In Borck-Pereyra algorithm, the variable S_i which initially contains the i th syndrome value is updated iteratively. Instead of using β - matrix to compute (2), Borck-Pereyra algorithm uses β - vector to reduce the implementation complexity. After all computations, S_i indicates the i -th error magnitude. From Table 5, Borck-Pereyra algorithm has division, multiplication and addition operations. Notice that

the multiplier can be shared if the divider can be decomposed into an inversion and a multiplier. Thus, as shown in Figure 15, BPEMS only contains 1 multiplier, 1 inversion, 3 adders and a control logic. The control logic determines the computation order of the syndromes and β_{ci} , and the computation results will be used to update each S_i value. The inversion in the proposed architecture is carried out in composite field because the finite field inversion over $GF(2^m)$ is costly and infeasible with table-lookup implementation for large m .

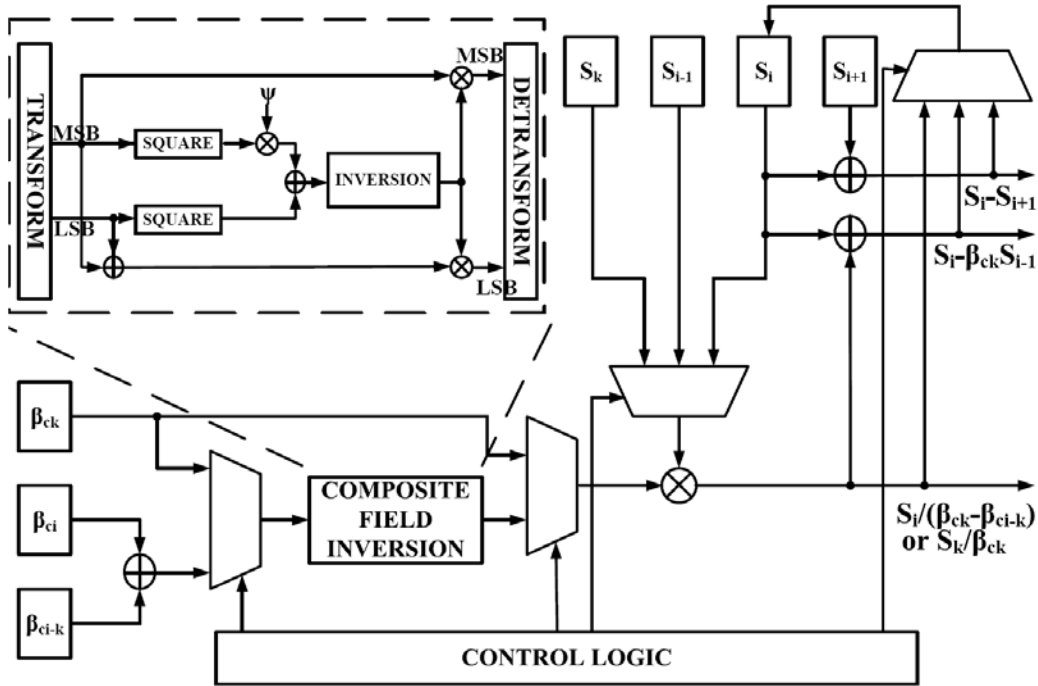


FIGURE 15: BORCK-PEREYRA ERROR MAGNITUDES SOLVER ARCHITECTURE

Composite field [B-11] is viewed as an extension field of $GF(2^k)$ while given $m = kr$. The finite field $GF(2^m)$ can be constructed by coefficients from the subfield $GF(2^k)$. Operating in subfield leads to lower implementation complexity and better computation efficiency. For example, every element in $GF(2^{16})$ can be represented by $bx+c$ and inversion of $bx+c$ can be derived as (4) with the polynomial $x^2 + x + \phi$ [B-11], where b and c are over $GF(2^8)$.

$$\frac{1}{bx + c} = (b^2\psi + bc + c^2)^{-1}(bx + b + c) \quad (4)$$

The composite field inversion over $GF(2^{16})$ is only 2.1K gate count in CMOS 90nm technology while the inversion using Look Up Table method is about 186K gate count.

3. Simulation and Implementation Result in DMB-T and DVB-S2 Systems

In DMB-T and DVB-S2 systems, BCH (762,752) over $GF(2^{10})$ and BCH (32400,32208) over $GF(2^{16})$ are defined to be concatenated with LDPC codes respectively. Figure16 shows the simulation results for DMB-T system with LDPC (7493,3048) at 20 iterations. Similarly, the simulation results for DVB-S2 system with LDPC (64800,32400) at 50 iterations is shown in Figure17. The proposed soft BCH decoders have 0.5db gain in DMB-T and similar performance in DVB-S2 at $BER = 10^{-5}$. These two BCH codes are implemented and demonstrated in Table 6. Each BCH code is implemented in both hard decision and soft decision methods.

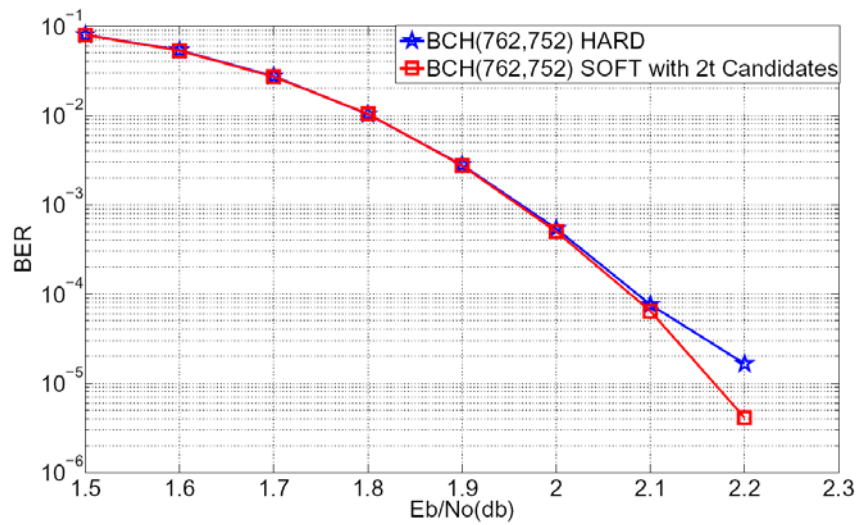


FIGURE16: SIMULATION RESULTS FOR BCH (762,752,1) IN DMB-T SYSTEM UNDER BPSK MODULATION AND AWGN CHANNEL

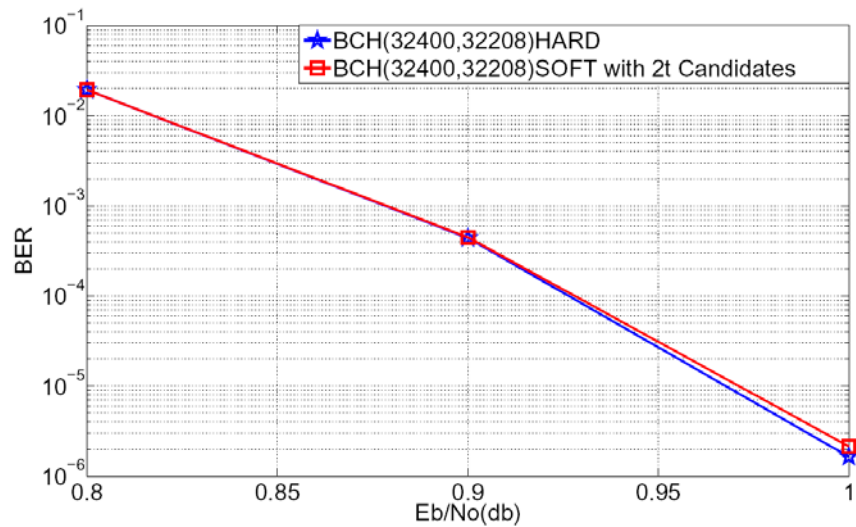


FIGURE 17: SIMULATION RESULTS FOR BCH (32400,32208,12) IN DVB-S2 SYSTEM UNDER QPSK MODULATION AND AWGN CHANNEL

TABLE 6: SUMMARY OF IMPLEMENTATION RESULTS

	Hard BCH (762,752), t = 1	Soft BCH (762,752), t = 1	Hard BCH (32400,32208), t = 12	Soft BCH (32400,32208), t = 12
Technology	90nm	90nm	90nm	90nm
Architecture	Look Up Table	H-EMS	iBM + Chien Search	BP-EMS
Operation Frequency	500MHz (Post Layout)	500MHz (Post Layout)	200MHz (Post Layout)	333MHz (Measurement)
Core Area	14400 μm^2	4225 μm^2	190497 μm^2	102400 μm^2
Gate Count	3.1K	1.2K	54.0K	26.9K
Latency	763	763	64824	34104
Throughput	492.8Mbps	492.8Mbps	99.3Mbps	314.5Mbps

For BCH (762,752), key equation procedure is not needed due to $t = 1$. To eliminate Chien search, the hard BCH decoder uses look up table method to solve the error location, and the soft BCH decoder uses the H-EMS architecture. Calculating all the combination values at one cycle, the gate-count of the soft BCH decoder is only 38.8% of the hard BCH decoder under the same latency and operation frequency. For BCH (32400,32208) with $t = 12$, the hard BCH decoder uses iBM algorithm to solve key equation and needs Chien search to get error locations. By inserting registers in composite field inversion, the operation frequency of the soft BCH decoder with BP-EMS is enhanced from 166MHz to 333MHz with only 2.5% latency increment in overall decoding procedure. Computing error locations without Chien search, the soft BCH decoder has almost half latencies of the hard BCH decoder. Hence, the soft BCH decoder has much better throughputs than the hard BCH decoder. The measurement result reveals that the soft BCH decoder saves 50.0% gate-count and 47.4% clock cycle latency as compared with the hard BCH decoder. Figure 18 is the chip microphoto of soft BCH (32400,32208).

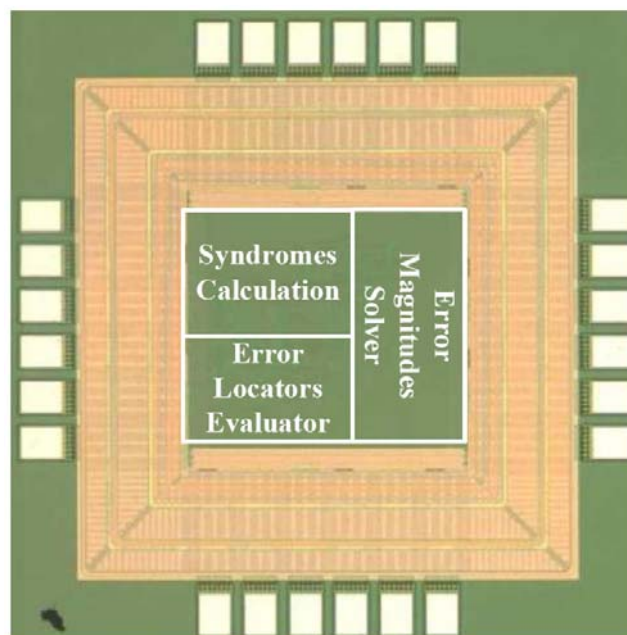


FIGURE 18: MICROPHOTO OF SOFT BCH(32400,32208) CHIP

四、 結論與討論

Among this year, we propose two modified FEC decoders for this low-power base-band processor and describe those contributions as follows:

We present a low-power Viterbi decoder combining the SST and the variable truncation length scheme. Reducing the state transition activities, the SST approach reduces the dynamic power in the decoder. The variable truncation length scheme provides a more efficient survivor memory management that pursues the sufficient truncation lengths for real channel conditions. Consequently, the redundant data movement can be abandoned for low power. Furthermore, high SNR environment causes the survivors merge rapidly, leading to more gated register elements. After implemented with the 0.13- μm cell based design flow, the fix-point error performance, the gate count, and the power dissipation has been examined by applying the presented low power techniques. Experimental results indicate the power reduction of the whole decoder and the survivor memory unit can achieve more than 30% and 60% respectively, while the overhead of 8% gate count due to additional control logics is required.

We also provide soft BCH decoders suitable for digital video broadcasting. From the simulation, the proposed soft decoders perform much lower complexity with similar system performance as compared with conventional hard decoders. The proposed soft BCH decoders can save 61.2% gate-count for BCH (762,752) with H-EMS to correct 1 error. For BCH (32400,32208) with BPEMS, which can correct 12 errors, 50.0% gate-count and 47.4% clock cycle latency are saved compared with traditional hard BCH decoders in CMOS 90nm technology.

五、 參考文獻

- [A-1] A. J. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm," IEEE Trans. Inform. Theory, vol. IT-13, no. 2, pp. 260-269, Apr. 1967.
- [A-2] G. Feygin and P. G. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," IEEE Transactions on Communications, vol. 41, no. 3, pp. 425-429, March, 1993.
- [A-3] C. C. Wu, "A High-Performance and Low-Power Viterbi Decoder," MS. thesis, National Chiao Tung Univ., 2003.
- [A-4] A. Batra et al., "Multi-band OFDM physical layer proposal for IEEE 802.15 task group 3a," submitted to IEEE P802.15 working group for WPANs, Sept. 2004.
- [A-5] P. J. Black and T. H. Meng, "A 140-Nb/s, 32-state, radix-4, Viterbi Decoder," IEEE J. Solid-State Circuits, vol. 27, no. 12, pp. 1877-1885, Dec. 1992.
- [A-6] S. Kubota and S. Kato, "Novel Viterbi Decoder VLSI Implementation and its Performance," IEEE Trans. Commun., vol. 41, no. 8, pp. 1170-1178, Aug. 1993.
- [A-7] L. H. C. Lee, D. J. Tait, and P. G. Farrell, "Scarce-State-Transition Syndrome-Former Error-Trellis Decoding of $(n,n-1)$ Convolutional Codes," IEEE Trans. Commun., vol. 44, no.1, pp. 7-9, Jan. 1996.
- [A-8] S. Kubota, K. Ohtani, and S. Kato, "High-Speed and High-Coding-Gain Viterbi Decoder with Low Power Consumption Employing SST (Scarce State Transition) Scheme," Electron. Lett., vol. 22, no.9, pp. 491-493, Apr. 1986.
- [A-9] C. C. Lin, Y. H. Shih, H. C. Chang, and C. Y. Lee, "Design of a Power-Reduction Viterbi Decoder for WLAN Applications," IEEE Trans. Circuit and Syst. I, vol. 52, no. 6, pp. 1148-1156, June 2005.
- [B-1] C. R. Baugh and B. A. Wooley, Theory and Practice of Error Control Codes. Addison-Wesley, 1983.
- [B-2] Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System, NSPRC Std. GB 20 600-2006, 2007.
- [B-3] Digital Video Bracasting (DVB) Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications, ETSI Std. En 302 307, 2005.
- [B-4] G. D. Forney, "Generalized Minimum Distance Decoding," IEEE Trans. Inform. Theory, vol. 12, p. 125V131, Apr. 1966.
- [B-5] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," IEEE Trans. Inform. Theory, vol. IT-18, p. 170V182, Jan. 1972.
- [B-6] M. Lalam, K. .Amis, D. Lerous, D. Feng, and J. Yuan, "An Improved Iterative

Decoding Algorithm for Block Turbo Codes,” IEEE Int. Symp. on Info. Theory, pp. 2403–2407, July 2006.

[B-7] W. J. ReidIII, L. L. Joiner, and J. J. Komo, “Soft Decision Decoding of BCH Codes Using Error Magnitudes,” IEEE Int. Symp. on Info. Theory, p. 303, June 1997.

[B-8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” IEEE Trans. Inform. Theory, vol. 20, pp. 284–287, Mar. 1974.

[B-9] A. Björck and V. Pereyra, “Solution of Vandermonde Systems of Equations,” Math. Computation, vol. 24, pp. 893–903, Oct. 1970.

[B-10] J. Hong and M. Vetterli, “Simple Algorithms for BCH Decoding,” IEEE Trans. on Communication, vol. 43, pp. 2324–2333, Aug. 1995.

[B-11] C. Parr, “Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields,” Ph.D. dissertation, Inst. for Experimental Mathematics of Univ. of Essen Germany, 1994.

六、計畫成果自評

在此計畫執行第一年中，我們提供兩個可應用於無線通訊之低功耗基頻處理器元件，其中分別為：

- 1) A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length
- 2) Soft BCH Decoder Using Error Magnitudes for Digital Video Broadcasting

下表七與八是為本研究團隊對於今年執行的計畫進度與各別研究規劃，在表中，我們完成了此兩研究的架構設計、下線與量測；在此計畫第一年的研究成果，在未來，希望能為產業界、學術界盡一份心力。

表七 本年度(2008)子計畫一之研究規劃

月份 (2008)	01	02	03	04	05	06	07	08	09	10	11	12
A Low-Power Viterbi Decoder Based on Scarce State Transition and Variable Truncation Length												
Modified SST Algorithm												
Modified SST Architecture Design												
Modified SST VD FPGA Platform												
Modified SMU Prepare												
Modified SMU Design & Simulation												
Modified SST + SMU VD FPGA Platform												
Modified SST + SMU VD Chip Layout												
Modified SST + SMU VD IC Tape-out												

表八 本年度(2008)子計畫二之研究規劃

月份 (2008)	01	02	03	04	05	06	07	08	09	10	11	12
Soft BCH Decoders Using Error Magnitudes for Digital Video Broadcasting												
Soft BCH Paper Survey												
Soft BCH Algorithm Simulation for DMB-T												
Soft BCH Algorithm Simulation for DVB-S2												
Soft BCH Architecture Design												
Soft BCH for DMB-T APR												
Soft BCH for DVB-S2 Tape-out												

表九簡列此計畫支持本研究群的相關研究成果，其中，有 9 件國內、外專利申請中、22 篇國際期刊與會議論文已發表於 IEEE，本研究群亦十分感謝國家科學委員會今年暨未來的持續支持與鼓勵。

表九 本計畫近年(2007-2009)研究貢獻

成果項目			96.01.01- 98.05.31	成果項目			96.01.01- 98.05.31
專 利	申請	國內 (件數)	4	論 文	期刊	國內 (件數)	0
		國外 (件數)	5			國外 (件數)	9
		國內外合計件數	9			國內外合計件數	9
	獲得	國內 (件數)	0		研討會	國內 (件數)	0
		國外 (件數)	0			國外 (件數)	13
		國內外合計件數	0			國內外合計件數	13