EFFICIENT COHERENT DETECTOR VLSI DESIGN FOR CONTINUOUS PHASE MODULATION

Tong Zhang, Jie Wu and Gary J. Saulnier

Electrical, Computer and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, NY, USA
Email: tzhang@ecse.rpi.edu, wuj2@rpi.edu, saulng@rpi.edu

ABSTRACT

This paper presents a continuous phase modulation (CPM) coherent detector design suitable for high throughput, low power VLSI implementations. The key component of CPM coherent detector is the trellis decoder. Compared with the Viterbi algorithm, the sub-optimum T-algorithm significantly reduces computation complexity and provides great potential to realize low power trellis decoding. But it is not suitable for VLSI implementations because it contains search-the-best-metric operation as a throughput bottleneck and suffers from unstructured data manipulation. We propose an algorithm level technique to eliminate the throughput bottleneck in the original T-algorithm, leading to a new SPEC-T algorithm. A VLSI architecture for implementing the SPEC-T algorithm is developed, in which a token bus structure is used to solve the unstructured data manipulation problem.

1. INTRODUCTION

In this work, we are interested in a high throughput, low power VLSI implementation of a continuous phase modulation (CPM) [1] coherent detector. CPM has received much attention due to its spectral efficiency and constant envelope property. It is well known that the optimum coherent detector using the Viterbi algorithm (VA) suffers from high computation complexity and hence high power consumption because the spectral efficiency of CPM comes with a large number of trellis states. Nevertheless, due to its highly regular operation, VA is well suited to a high throughput VLSI implementation.

Sub-optimum detection based on *reduced search* algorithms, i.e., M-algorithm [2] and T-algorithm [3], achieves near-optimum performance with much less computation complexity, thus have great potential to reduce the CPM detector power consumption. However, the M- and T- algorithms are not suitable for high throughput VLSI implementations for two main reasons: 1. the *sort-the-metric* (in M-algorithm) and *search-the-best-metric* (in T-algorithm) operations in each trellis decoding step are essentially serial and remain as a critical throughput bottleneck, and 2. unstructured data manipulation of survivor path seriously complicates the parallel data storage and retrieval mechanism design and significantly decreases the throughput and increase the power consumption.

In this paper, we propose two techniques at the algorithm and architecture levels to attack the above two problems for sub-optimum CPM coherent detector using the T-algorithm. We first propose a speculation technique to move the search-the-best-metric operation out from the main recursive detection data-path to increase the throughput. This leads to a *SPEC-T* algorithm. For the architecture design of SPEC-T algorithm, we develop a survivor path re-distribution scheme based on a token bus to effectively solve the parallel data storage and retrieval problem. Compared with the CPM detector employing a state-parallel Viterbi decoder, this proposed SPEC-T CPM detector achieves the comparable CPM signal detection performance and throughput while consuming much (even an order of magnitude) less energy consumption and silicon area.

2. BACKGROUND

In this section, we briefly review some basics of CPM and the coherent detector using T-algorithm. A CPM signal can be described as

$$s(t) = e^{j\phi(t)},$$

where

$$\phi(t) = 2\pi h \sum_{n=0}^{k} \alpha_n q(t - nT_p), kT_p < t < (k+1)T_p.$$

The $\{\alpha_n\}$ are M-ary data symbols, T_p is the symbol period, $h = \frac{2k_l}{p}$ is modulation index, in which k_l and p are relatively prime, and $q(t)$ is the phase pulse function of length

L and is a continuous, monotonic function with the restriction

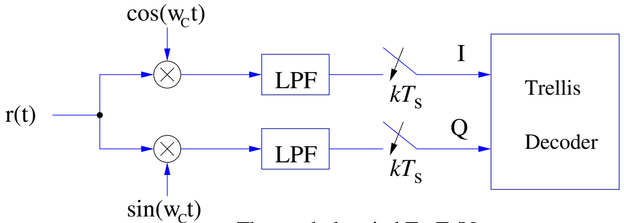
$$q(t) = \begin{cases} 0, & t \leq 0 \\ \frac{1}{2}, & t \geq LT_p \end{cases} .$$

The optimum coherent detector [1] demands pM^L matched filters followed by a Viterbi decoder with pM^{L-1} states, where each matched filter calculates a distinct trellis branch metric. In this work, instead of using a costly matched-filter bank to obtain the optimum trellis branch metric, we adopt quadrature demodulation followed by oversampling to move the trellis branch metric calculation into the digital domain for efficient implementation [4] as shown in Fig. 1. The T-algorithm for CPM signal detection can be described as follows:

1. ***Path Extension***: Extend the survivors from the previous depth. With the oversampling factor of N, we have N I-Q inputs per symbol: $(I_1^{(n)}, Q_1^{(n)}), \cdots, (I_N^{(n)}, Q_N^{(n)})$. Each trellis branch corresponds to N ideal noiseless CPM signal samples $e^{j\phi_1}, \cdots, e^{j\phi_N}$. The branch metric is $\sum_{i=1}^{N} A_i^{(n)} \cos(\phi_i - \hat{\phi}_i^{(n)})$, where $A_i^{(n)} e^{j\hat{\phi}_i^{(n)}} = I_i^{(n)} + jQ_i^{(n)}$.

2. ***Best Metric Search***: Find the contender path having the best (largest) path metric $\Gamma_B^{(n)}$ and release its depth-S symbol $\hat{\alpha}_{n-S+1}$ to the output.

3. ***Path Purge***: (a) Purge all the contender paths whose path metric $\Gamma^{(n)}$ satisfies $\Gamma_B^{(n)} - \Gamma^{(n)} > T$, where T is a pre-specified threshold, and (b) purge all the contender paths that do not agree with $\hat{\alpha}_{n-S+1}$.



The symbol period Ts=Tp/N

Fig. 1. The principal structure of the coherent detector.

3. PROPOSED CPM DETECTOR

We propose a SPEC-T trellis decoding algorithm and its VLSI architecture design for high throughput, low power implementation of a CPM coherent detector. In the original T-algorithm, as remarked in Fig. 2 (a), the *Branch Metric Computation & Path Extension* and *Path Purge* can be carried out in parallel among all the survivor paths. However, the *Best Metric Search* obviously incurs large delay due to the serial essence of the search-the-best-metric operation, which becomes a throughput bottleneck.

3.1. SPEC-T Algorithm Design

To eliminate such inherent decoding throughput bottleneck, we propose a *speculation* technique to modify the original T-algorithm, leading to a SPEC-T algorithm. The basic idea is *best metric speculation with lagged correction*: instead of searching for the *exact* best metric at each depth, we *speculate* the best metric based on the current I-Q input and perform an *off-the-main-recursion* search to regularly correct the speculation error with a certain delay. As shown in Fig. 2 (b), this SPEC-T algorithm contains two function modules:

1. *Detection Module* that performs the branch metric calculation, path extension, and path purge by speculating the best metric, and

2. *Correction Module* that, every v detection depths, adjusts the best metric speculation to remove the accumulated speculation error and releases v output symbols.

The operations performed by these two modules are outlined in the following.

Detection Module:

1. ***Branch Metric Computation & Path Extension***: same as the original T-algorithm.

2. ***Best Metric Speculation***: At depth n, speculate the best metric $\hat{\Gamma}_B^{(n)}$ as follows:

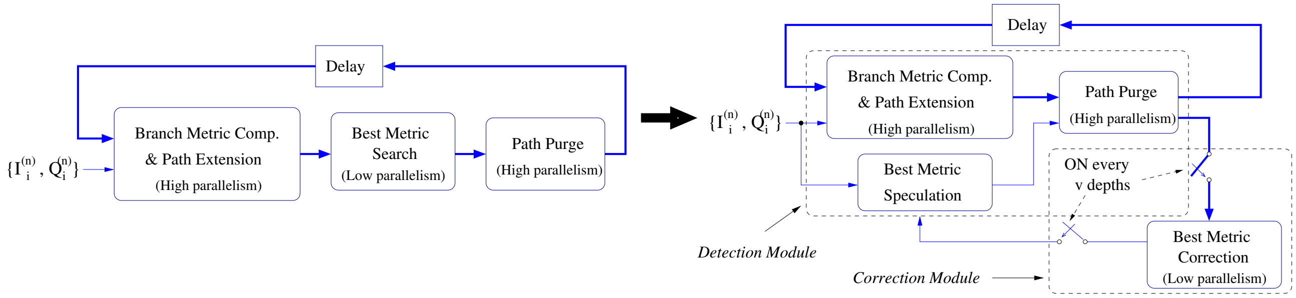
   - If $n \bmod v \neq 0$, then $\hat{\Gamma}_B^{(n)} = \hat{\Gamma}_B^{(n-1)} + \sum_{i=1}^{N} A_i^{(n)}$, where $A_i^{(n)}$ is the magnitude of the I-Q sample $I_i^{(n)} + jQ_i^{(n)}$.
   - If $n \bmod v = 0$, then $\hat{\Gamma}_B^{(n)} = \hat{\Gamma}_B^{(n-1)} + \sum_{i=1}^{N} A_i^{(n)} - E^{(n-v)}$, where $E^{(n-v)}$ is provided by the Correction Module to compensate the accumulated speculation error.

3. ***Path Purge***: same as the original T-algorithm, and when $n \bmod v = 0$, send the metric difference $\hat{\Gamma}_B^{(n)} - \Gamma_i^{(n)}$ and v oldest path symbols of all the contender paths to the Correction Module.

Correction Module:

1. Search for $E^{(n-v)} = \min\{\hat{\Gamma}_B^{(n-v)} - \Gamma_i^{(n-v)}\}$.

2. Release the v symbols, $\hat{\alpha}_{n-L-2v+1}, \cdots, \hat{\alpha}_{n-L-v}$, associated with the overall best path leading to $E^{(n-v)}$.


Delay
$\{I_i^{(n)}, Q_i^{(n)}\}$
Branch Metric Comp. & Path Extension (High parallelism)
Best Metric Search (Low parallelism)
Path Purge (High parallelism)
Delay
$\{I_i^{(n)}, Q_i^{(n)}\}$
Branch Metric Comp. & Path Extension (High parallelism)
Path Purge (High parallelism)
Best Metric Speculation
ON every v depths
Best Metric Correction (Low parallelism)
Detection Module
Correction Module

(a) The original T–algorithm

(b) The high throughput SPEC–T–algorithm

Fig. 2. The T-algorithm modification for high throughput.

Notice that the iteration of Detection Module runs v times faster than that of Correction Module, which can compensate the unmatched delay between the high-parallelism operations in Detection Module and the low-parallelism of the *search-the-best-metric* operation in Correction Module. Thus, by moving the inherently serial searching operation out from the main recursive decoding data-path, we can completely exploit the parallelism of the branch extension and path purge to significantly speed up the decoding. However, due to the lagged correction of the speculation error, this will result in CPM signal detection performance degradation, which is the price we have to pay for the decoding speed-up. The question is whether such degradation is tolerable for the specific applications. As we will show later, the performance degradation is generally very small or even negligible.

3.2. Detector Structure Design

For the practical implementation of the SPEC-T algorithm, we introduce two parameters: M_{max} and M_{min}, the maximum and minimum number of the survivors retained after each detection depth. If the number of survivors obtained under the threshold T is larger (less) than M_{max} (M_{min}), then we reduce (increase) T by 10% and repeat the current detection depth until the number falls between M_{min} and M_{max}.

Fig. 3 shows the SPEC-T algorithm decoder VLSI architecture. Each processing element PE_i performs the branch metric calculation and path extension for one survivor and path purge for its extended paths. Each register array PD_i stores the corresponding path information. At the end of each depth, each PD_i should contain the path information of at most one survivor in order to enable the parallel processing in the next detection depth. This demands *moving* some path information among the PD_i's, which leads to the data storage and retrieval problem as mentioned above. We develop the following *token bus based data re-distribution* mechanism to attack this problem.

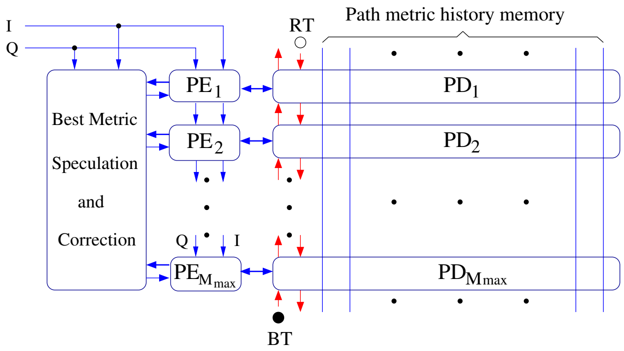

I
Q
RT
Path metric history memory
Best Metric Speculation and Correction
PE$_1$
PE$_2$
PE$_{M_{max}}$
Q
I
PD$_1$
PD$_2$
PD$_{M_{max}}$
BT

Fig. 3. The general architecture of SPEC-T decoder.

Notice that after the path purge, we may categorize the M_{max} PD_i's as follows: 1. empty register array PD_i^E that does not contain any survivor, 2. carefree register array PD_i^C that contains one survivor, and 3. congested register array PD_i^G that contains more than one survivor. We propose to use a *token bus* to move the extra path information in the congested register arrays to empty register arrays in high speed and without incurring too much complexity and power overhead. As shown in Fig. 3, after the path purge, the detector initiates two tokens, one is the *broadcasting* token BT and another is the *receiving* token RT, and we have

- The carefree register array PD_i^C simply bypasses both the BT and RT.
- The empty register array PD_i^E receives the RT and bypasses the BT.
- The congested register array PD_i^G receives the BT and bypasses the RT.

Each clock cycle, one PD_i^G and one PD_i^E hold the BT and RT, respectively. The PD_i^G that acquires BT *broadcasts* one survivor path information to the bus, which is received by the PD_i^C that acquires RT. We note that because of the simple operations involved in the data re-distribution,

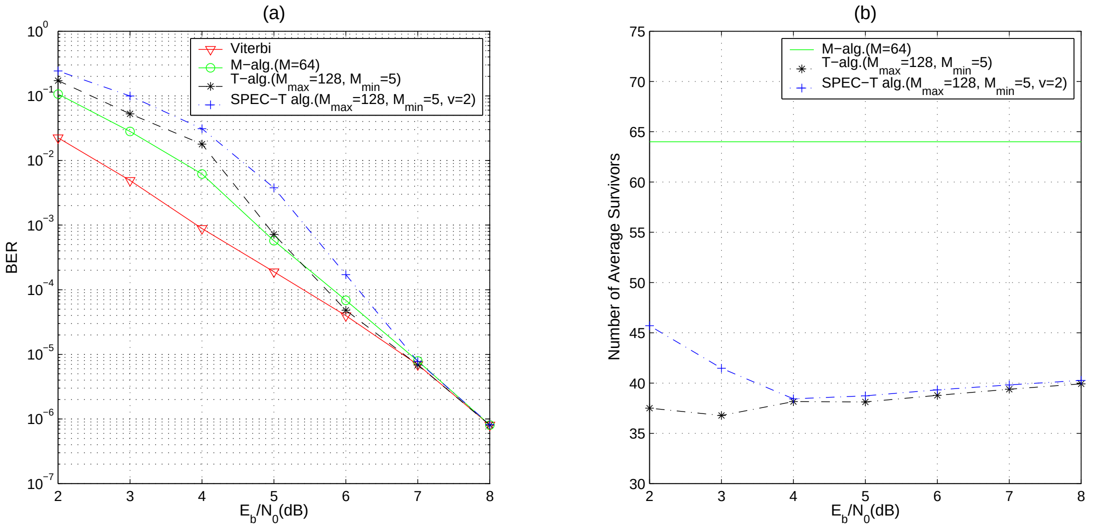


Fig. 4. Simulation results of (a) BER vs. SNR, and (b) average number of survivors vs. SNR.

the clock that controls the data re-distribution can run much faster than the clock that controls the operation of PE_i's. Thus the overall delay incurred by the data re-distribution is not significant.

3.3. Simulation Results

Consider the coherent detection of quaternary $h = 2/5$ 5RC CPM over an AWGN channel. Its trellis contains $5 \times 4^4 = 1280$ states. Fig. 4 shows the simulated bit error rate (BER) using the VA, M-, T-, and SPEC-T algorithms, and the average number of survivors for T- and SPEC-T algorithms. Gray mapping is used for the CPM signaling. The oversampling factor is 2 for all the cases. The performance degradation of using SPEC-T algorithm is negligible at BER of 10^{-5}. For the SPEC-T algorithm, in Table 1, we present the simulation results showing two parameters pertaining to the detection throughput: 1. DL_o: the average number of detection depths repeated every 1K symbols due to the adjustment of threshold T, and 2. NC_r: the average number of the broadcasting-receiving operations to complete the data re-distribution at each detection depth.

SNR	2 dB	3 dB	4 dB	5 dB	6 dB	7 dB
DL_o	394	213	97	49	33	23
NC_r	23.8	17.3	12.9	11.6	11.4	11.2

Table 1. The simulation results of the two parameters pertaining to the detection throughput.

3.4. Conclusions

This paper presented an algorithm/architecture design solution for high throughput, low power CPM coherent detector VLSI implementations. At algorithm level, a speculation based technique is proposed to eliminate the inherent throughput bottleneck of the T-algorithm, leading to SPEC-T algorithm. Consequently, we develop a decoder architecture for SPEC-T algorithm, in which we apply the concept of token bus to solve the unstructured data storage and retrieval problem. This proposed design provides the communication system designers an unique opportunity to exploit the attributes of the T-algorithm, i.e., low computational complexity and near-optimum performance, to significantly reduce the CPM detector implementation complexity and power consumption while achieving high throughput.

4. REFERENCES

[1] J. B. Anderson and A. Svensson, *Coded Modulation Systems*, Kluwer Academic / Plenum Publishers, 2003.

[2] J. B. Anderson and S. Mohan, “Sequential coding algorithms: A survey and cost analysis,” *IEEE Transactions on Communications*, vol. 32, pp. 169–176, Feb. 1984.

[3] S. J. Simmons, “Breadth-first trellis decoding with adaptive effort,” *IEEE Transactions on Communications*, vol. 38, pp. 3–12, Jan. 1990.

[4] S. J. Simmons, “Simplified coherent detection of CPM,” *IEEE Transactions on Communications*, vol. 43, pp. 726–728, Feb/Mar/Apr 1995.