

# An Adaptive Viterbi Decoder on the Dynamically Reconfigurable Processor

Shohei Abe<sup>†</sup>, Yohei Hasegawa<sup>†</sup>, Takao Toi<sup>‡</sup>, Takeshi Inuo<sup>‡</sup>, and Hideharu Amano<sup>†</sup>

<sup>†</sup> *Department of Information and Computer Science, Keio University  
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama 223-8522, Japan*

<sup>‡</sup> *NEC System Devices Research Labs.  
1753 Shimonumabe, Nakahara-ku, Kawasaki 211-8668, Japan*

drp@am.ics.keio.ac.jp

**Abstract**—In order to evaluate practical adaptive computing on dynamically reconfigurable processors, several Viterbi decoders with different constraint variables are implemented on NEC Electronics' DRP-1. By switching designs, its throughput varies from 4.71 Mbps to 9.95 Mbps and its power consumption does from 423.93 mW to 1028.97 mW at the fixed throughput in response to the Signal to Noise Ratio. The power can be saved up to 58.3% and the throughput can be improved 2.1 times by switching designs appropriately when the distance of the base station and the mobile terminal is not very long.

## I. INTRODUCTION

Mobile computing systems, powered for hours or days by batteries, are required to suppress energy consumption, maintaining quality of service. Especially in mobile communication, it is required to maintain a certain bit error rate (BER) while signal-to-noise (SNR) on a channel varies. The system should optimize its error correcting capability to save energy in response to the variation of SNR.

*Adaptive computing*[1][3] is useful technique for power saving by making full advantage of reconfigurable devices. In this method, the system uses a set of multiple configuration data which provides the same function with the different performance and power consumption. In response to a condition, a reconfigurable system or device changes its configuration so as to fit the given condition and improve performance and energy-efficiency.

The effectiveness of the adaptive computing depends on the device in energy and time for reconfiguration especially. Although the adaptive computing on traditional FPGA has been reported in [1], it spends a couple dozen milliseconds to change its hardware structure and extra power consumption for loading configuration data. The long configuration time of traditional FPGAs leads to a temporal growth of the latency which can be fatal in communication.

On the contrary, various types of coarse grain dynamically reconfigurable devices have been announced in recent few

years[4][5][2], and some of them have been used in a portable game engine, network controllers[5] and intelligent digital video cameras. Since they can change their configuration dynamically, the configuration time and energy are much smaller than those of FPGAs. Some of these devices employ a multicontext structure that provides a set of configuration memory modules in each of the processing elements. By broadcasting the pointer to the individual configuration memories, the hardware configuration can be changed in cycle-by-cycle basis. Hardware configuration is usually referred to as a 'hardware context', and these contexts are interchanged in a clock cycle if needed. By using such devices, the barrier for introducing adaptive computing is much reduced.

In this paper, we have implemented five Viterbi decoders with different parameters on the NEC Electronics' DRP-1, the first silicon implementation of the DRP architecture. We have applied the method to adaptive computation on mobile systems in a simulation model, and analyze the efficiency to save the power consumption.

## II. DRP-1 OVERVIEW

The DRP-1 is a coarse-grain multi-context reconfigurable device. It has configuration sets corresponding to contexts. One context defines just one hardware structure pattern. The DRP-1 has sixteen contexts. As shown in Fig 1, the DRP-1 consists of a memory controller, a PCI controller, eight 32-bit multipliers and 4×2 individually reconfigurable units, each of which is called a Tile.

Each Tile includes 8×8 Processing Element (PE) array, a State Transition Controller (STC), eight 2-ported memories (VMEMs: Vertical MEMories), two VMEM Controllers (VMCTR), four 1-ported memories (HMEMs: Horizontal MEMories), and one HMEM Controller (HMCTR). Each PE has an 8-bit ALU, a Data Management Unit, a flip flop, and a register file. All units in the PE treat 1-bit and 8-bit operations, and the function of each PE can be reconfigured every clock.

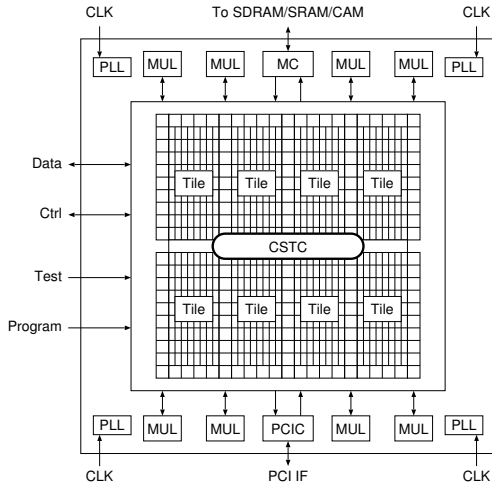


Fig. 1. DRP-1 Architecture

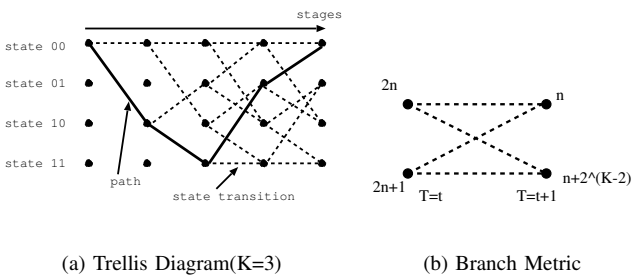


Fig. 2. Trellis diagram

### III. RELATED WORK

Adaptive computing using traditional FPGAs has been tried on traditional FPGAs. In [1], an adaptive Viterbi decoder is designed for optimization of power consumption. Although some effects are reported, the decoder must stop during the reconfiguration and the extra power consumption for loading configuration data is required. The reduction of power consumption through dynamic reconfiguration for SRAM FPGA was analyzed by general model in [10], and it results that the power for reconfiguration itself sometimes ruins the effect of dynamic configuration. Some trials for adaptive computing using DRP-1 have been reported[3][9]. However, both trials focus on only performance improvement.

### IV. VITERBI DECODER

The behavior of a Viterbi decoder is often represented with the trellis diagram, as shown in Figure 2(a). Trellis is a kind of state transition diagram consisting of horizontal axis representing time (stage) and vertical axis for states which are formed with the input sequence. The number of state is  $2^K$  where  $K$  is the constraint length. The function of the decoder is to attempt to restrict the input sequence by making state transition patterns or *paths* from the received sequence. To

make a path, the decoder determines cost called *path metric* at each state. As shown in Figure 2(b), each state at  $T = t$  may transit to two states at  $T = t + 1$ , and the cost called *branch metric* is calculated according to the direction and input value of the decoder. The path metric for the next state is calculated from the sum of path metric in the current state and branch metric. The state transition sequence with the smallest path metric is selected as a path. In common Viterbi decoders, the above process is done with hardware module called ACS(Add-Compare-Sum) unit. The amount of required calculation in the ACS is increased with the number of  $K$ .

The original bit-sequence is decided from the selected path with the trace back unit. Instead of the LIFO based method used in the software implementation, systolic array[6] or Register Exchange Array(REA)[7] is commonly used in the hardware implementation. Although REA can make the best use of parallelism, it tends to require too large hardware resources for mobile devices. Here, the systolic array is used for trace back module.

In order to keep sufficient error correction capability, the length of the path(trace back length) should be  $5(K - 1)$ . In this case, the stage number of the pipelined the systolic array becomes  $5(K - 1)$ .

The size of the trellis diagram is decided with the constraint length  $K$ , that is, it becomes

$$\text{the number of states } 2^{K-1} \times \text{truncation length } 5K$$

[6]. The error-correcting capability of the convolution is improved by employing large size of trellis diagram with a large constraint length  $K$ .

On the other hand, the complexity of both the ACS and trace back module is increased with large  $K$ . Increasing  $K$  leads an exponential growth in the amount of computation and retained path storage. It also increases the power consumption and decreases the throughput.

Here, we focus on the constraint length  $K$  for introducing adaptive computing. Several decoder designs each of which is corresponding to different  $K$  are provided, and changed according to the S/N ratio. For severe condition with low degree of S/N ratio, a large sized design with a large  $K$  is used to require enough error correction capacity. When the S/N ratio is improved, that is, the condition of wireless communication becomes good, the power consumption can be saved by using the design of small  $K$ . That is, the decoder hardware can be dynamically switched so as to optimize the power consumption keeping the required error correction capacity.

### V. IMPLEMENTATION

We have implemented five Viterbi decoders with constraint length  $K = 3, 4, 5, 6, 7$  on the DRP-1. Other parameters

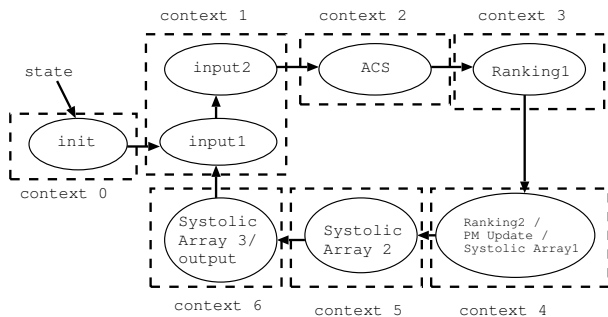


Fig. 3. Context Scheduling ( $K=7$ )

are set as follows: coding rate is  $1/2$ , trace back length is  $5(K-1)$ , and input/output width is 1 bit. The Viterbi decoding application was divided into states and allocated on contexts as shown in Figure 3. The figure is for  $K = 7$ , and those for smaller  $K$  are simpler than it. After data input in Context 1, the ACS is performed in Context 2. A decoder with constraint length  $K$  has  $2^{K-1}$  ACS Units, but all of them can be implemented on a context even when  $K = 7$ . Although the number of the context is the same, Context 2 for  $K = 7$  uses much more number of PEs than that for  $K = 3$ , and requires more power.

The trace back unit with constraint length  $K$  has  $5(K-1)$  pipeline stages for its systolic array structure. Since 30 stages are required for that for  $K = 7$ , they are divided and implemented on three contexts (Context 4, 5 and 6) as shown in Figure 3. The trace back unit for  $K = 6$  that requires 25 stages is implemented on two contexts, that is, the Context 5 in Figure 3 is omitted. In other designs with smaller  $K$  (3,4,5), every operation corresponding to Context 4, 5, and 6 can be mapped on only a context, that is, the required number of contexts can be reduced.

All designs are described in BDL: a C-like hardware description language, synthesized, and mapped on the DRP-1 with an integrated design tool called Musketeer.

## VI. EVALUATION

### A. Performance and Power Consumption

First of all, we investigated various features of the designed Viterbi decoders including error correction capability and power consumption. For this purpose, we implemented simulation programs on the host PC (Linux kernel 2.4.31, gcc 3.3.5 and boost 1.31.0 are used) as shown in Figure 4. The received bit sequences including a certain bit-errors are generated with the software on the host PC, transferred to the DRP-1 board, and decoded. The results are returned to the host PC and checked.

The measured results are shown in Table I. We note that power consumption shown in this table is estimated based on

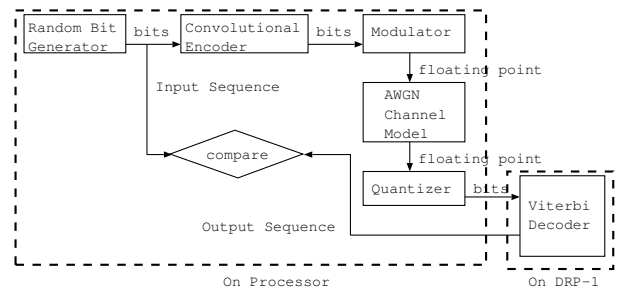


Fig. 4. The simulation model

TABLE I  
PERFORMANCE LIST

Constraint Length $K$	3	4	5	6	7
SNR ( $BER = 10^{-5}$ )	5.3	4.9	4.3	3.8	3.2
Num. of Contexts	2	4	4	5	7
Num. of Clocks (clocks/symbol)	4	5	5	6	7
Critical Path (ns)	25127	22340	26989	27390	30346
Max. Execution Freq. (MHz)	39.8	44.76	37.05	36.51	32.95
Power Consumption at Max. Freq. (mW)	910.52	1061.33	938.58	1025.06	1028.97
Throughput at Max. Freq.(Mbps)	9.95	8.95	7.41	6.08	4.71
Frequency at 4.71Mbps (MHz)	18.83	23.54	23.54	28.25	32.95
Power Consumption at 4.71Mbps (mW)	428.93	558.10	596.22	793.03	1028.97

the simulation. Hence, some errors may be observed in this estimation compared with the power consumption of real DRP chip.

The Viterbi decoder implemented on the DRP-1 with larger  $K$  has more powerful error-correcting capability, more contexts required, higher power consumption, and lower throughput than that with smaller  $K$ . This table shows that the maximum throughput is changed from 4.71Mbps ( $K = 7$ ) to 9.95Mbps ( $K = 3$ ).

For comparing the absolute performance, we implemented the same Viterbi coder on a common high-end embedded CPU: MIPS64 compatible VR5500. It is a 10-stage pipeline, 2-way out-of-order Super Scalar processor that runs at 400 MHz with 32 KB instruction cache and 32 KB data cache. The throughput of the decoder with  $K = 7$  on the VR5500 processor was 377 Kbits/symbol (10598 clocks/symbol), that is, much smaller than those of the Viterbi decoders on DRP-1.

Table I also shows the power consumption for achieving a fixed throughput 4.71Mbps. The power consumption with  $K = 7$  becomes more than double as that with  $K = 3$  under this condition. From this table, it appears that the power consumption can be saved by using designs with smaller  $K$ .

## B. Adaptive Computing on the DRP-1

We simulated mobile communication between a base station and a mobile terminal with 30dB, 40dB, and 50dB gain. Simulations of the adaptive computing were performed by varying the SNR of transmitted data and reconfiguring the ideal DRP based on  $K$  values that were required to achieve  $BER = 10^{-5}$ . The current DRP-1 provides 16 contexts, while the total contexts required for  $K = 3, 4, 5, 6$  and  $7$  becomes 23. In this evaluation, we assume the ideal case, that is, all 23 contexts can be stored in the context memory of the DRP-1, and the switching of the design can be done with a clock cycle.

The DRP device is reconfigured once every 250,000 symbols. A set of  $10^8$  SNRs was generated using a log-distance path loss model with path loss exponent  $n = 2.7$  and standard deviation  $\sigma = 11.8\text{dB}$  [8] for the total transmission length of  $250,000 \times 10^8$  bits<sup>1</sup>.

Figure 5 plots the variation in the power consumption at the fixed throughput (4.71 Mbps). The horizontal axis is corresponding to the distance between the base station and the mobile terminal. Without adaptive computing, since the design with  $K = 7$  is always used, and the power consumption is fixed independent from the distance. However, when the adaptive computing is introduced, the power can be saved when the distance is not so long, that is, the bit-error ratio is not too bad. When the distance is 0.6Km and the gain is 50dB, the power can be saved up to 58.3%.

Figure 6 shows the variation in the throughput depending on the distance. In this case, the operational frequency is changed when design is replaced so as to work at the maximum execution frequency of each design. Unlike the fixed throughput without the case of the adaptive computing, the throughput is also improved with the adaptive computing when the distance is not so long. When the distance is 0.6Km and the gain is 50dB, the throughput can be improved about 2.1 times.

## VII. CONCLUSION

Several Viterbi decoders with different constraint variables  $K$  are implemented on the dynamically reconfigurable processor DRP-1 to evaluate the effectiveness of practical adaptive computing. By switching the designs, its throughput varies from 4.71 Mbps to 9.95 Mbps and its power consumption does from 423.93 mW to 1028.97 mW at the fixed throughput in response to the Signal to Noise Ratio. The power consumption can be saved up to 58.3% and the throughput can be improved 2.1 times by switching designs appropriately when the distance of the base station and the mobile terminal is not so long.

## REFERENCES

- [1] R. Tessier, et al., "A Reconfigurable, Power-Efficient Adaptive Viterbi Decoder," in IEEE Transactions on VLSI Systems, vol.13, no 4, pp.484-488, April 2005.

<sup>1</sup>These parameters are based on the real measurement in German cities.

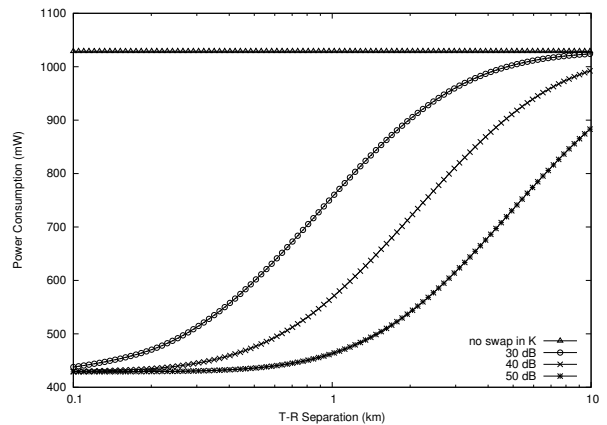


Fig. 5. Power Consumption vs. Distance (4.71Mbps)

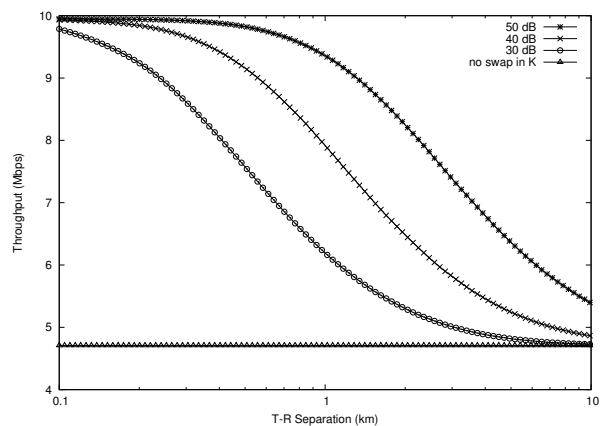


Fig. 6. Throughput at maximum exec. freq. vs. Distance

- [2] M. Motomura, "A Dynamically Reconfigurable Processor Architecture," Microprocessor Forum, October, 2002.
- [3] H. Amano, K. Anjo, and A. Jouraku, "A Dynamically Adaptive Hardware on Dynamically Reconfigurable Processor," IEICE Transactions, Vol.E86-B, No.12, pp.3385-3391, 2003.
- [4] F.-J. Veredas, M. Schepler, W. Moffat, B. Mei, "Custom Implementation of the Coarse-Grained Reconfigurable ADRES Architecture for Multimedia Purposes," Proc. of FPL2005, pp. 106-111, Sept. 2005.
- [5] T. Sugawara, K. Ide, T. Sato, "Dynamically Reconfigurable Processor Implemented with IPFlex's DAPDNA Technology," IEICE Trans. on Inf.&Syst. Vol.E87-D No.8, pp.1997-2003, May, (2004).
- [6] T.K. Truong, et al., "A VLSI Design for a Trace-Back Viterbi Decoder," in IEEE Transactions on Communications, VOL.40, NO. 3, March 1992.
- [7] E. Yeo, S. Augsburg, W.R. Davis, B. Nikolic, "Implementation of High Throughput Soft Output Viterbi Decoders," IEEE Workshop on Signal Processing Systems, Oct. 2002.
- [8] T. S. Rappaport, "Wireless Communications: Principles and Practice," Prentice Hall, Upper Saddle River, NJ, 1996.
- [9] Y. Hasegawa, et al., "An Adaptive Cryptographic Accelerator for IPsec on Dynamically Reconfigurable Processor," Proc. of ICFTPT05, pp.163-170, 2005.
- [10] M.G. Lorenz, L. Mengibar, M.G. Valderas and L. Entrena, "Power Consumption Reduction Through Dynamic Reconfiguration," Proc. of FPL 2004, pp.751-760, 2004.