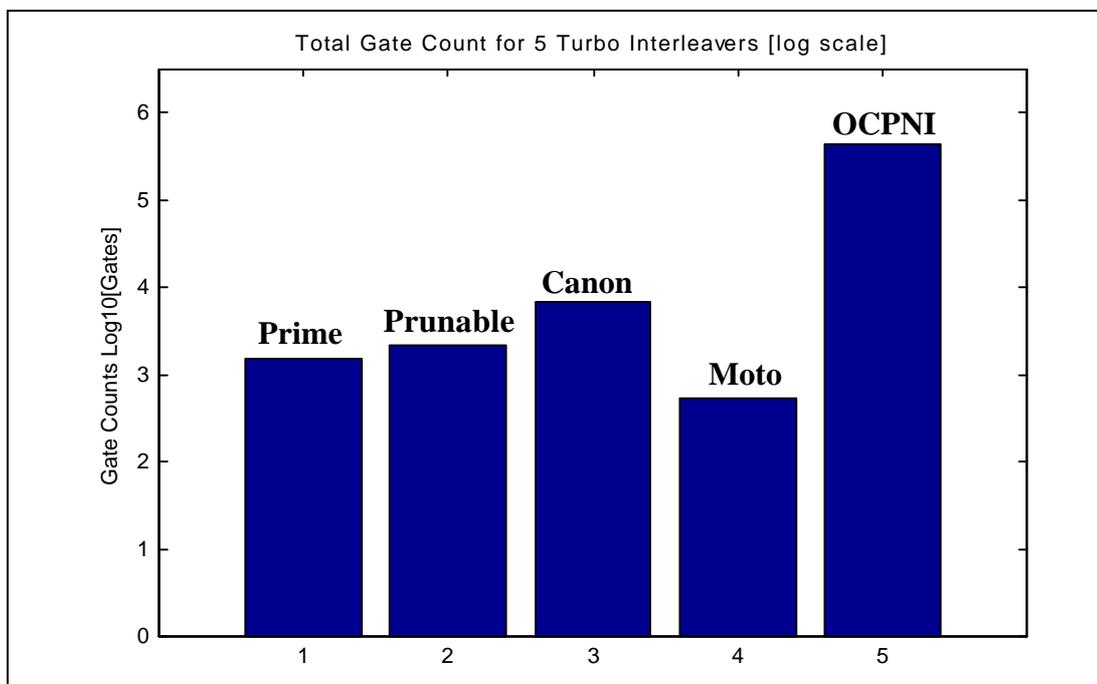


TABLE 6. Total Gate Counts for 5 Turbo Interleavers

Prime	Prunable	Canon	Motorola	OCPNI
2,629 gates	4,075 gates	12,892 gates	350 gates	690,508gates

FIGURE 5. Total Gate Counts Estimation



0.3.1 Relative Complexity to Turbo Decoder

The Log-Max-MAP decoder is used as benchmark for evaluation of Turbo Interleaver complexity. Two major parts of a Turbo decoder dominates the complexity (1) input buffer (2) likelihood function buffer which are implemented as RAM. For such a Turbo decoder operating at highest data rate, the RAM for input buffer and likelihood function buffer would be around 50kbits and 70kbits, i.e. 150kgates and 210kgates and 360kgates in total. Table 5 give the relative complexity of the Turbo interleaver w.r.t the complexity of Turbo decoder.

TABLE 7. Relative Complexity for 5 Turbo Interleavers

Prime	Prunable	Canon	Motorola	OCPNI
0.73%	1.113%	0.36%	0.1%	191%

As we can see the overall complexity for Turbo interrelaver relative to the Turbo decoder is insignificant except the OCPNI Interleaver.

FIGURE 3. ROM LUT Estimation

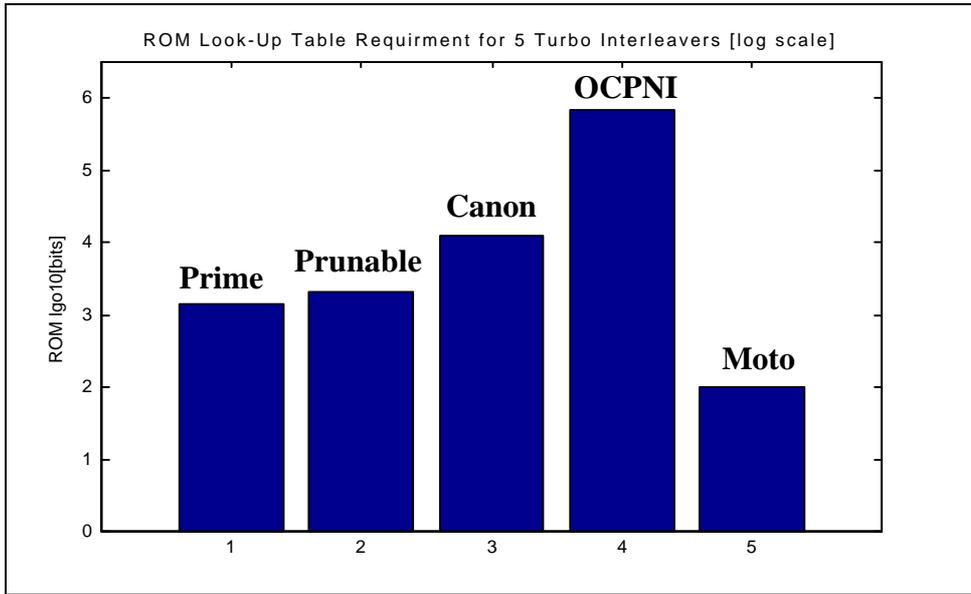
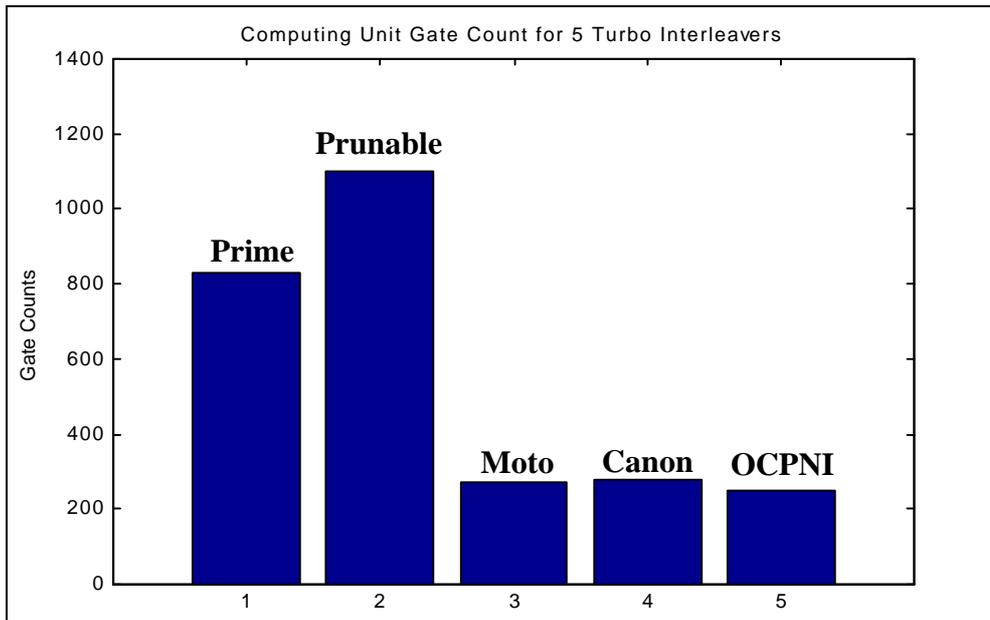


FIGURE 4. Gate Count Estimation



0.3 Overall Complexity Analysis of Turbo Interleavers

In order to evaluate the overall complexity of the Turbo interleaver, we assume an ASIC implementation for the entire Turbo decoder. Base on the general design rule-of-thumb 1 bit RAM equivalent to 3 gates and 1 bit ROM equivalent 0.5 gate, we obtain the overall complexity in terms of gate count for 4 Turbo interleavers as shown in Table 4.

The computing and control unit gate count are listed for 5 interleavers in Table 6.

TABLE 5. Gate Counts for 5 Turbo Interleavers

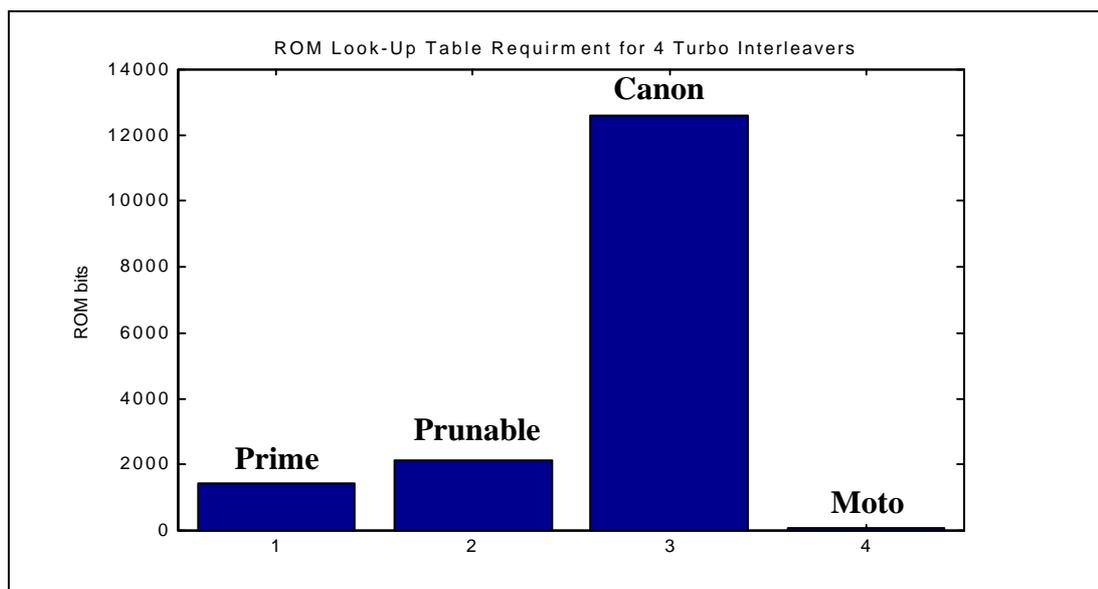
	Unit	Specification	Gates
Prime	9x9 bit multiplier	9-bit	450
	Glue Logic	Counter/Address Config.	60
	Total		290
Prunable	10x10 adder	10-bit	180
	Glue Logic	Counter/Bit-reversal/Addr.	60
	Total		240
Moto	Programmable LFSR	8-bit	150
	Glue Logic	Mux/Bit-reversal	80
	Total		230
Canon	13x13 adder-2	13-bit	200
	Glue Logic	Mux/bit-insertion	80
	Total		280
OCPNI	Programmable LFSR	5-bit/8-bit	270
	Glue Logic	Mux/Bit-reversal	80
	Total		350

Comments:

- To Exclude the common components such as comparator for address pruning.

As a comparison the ROM requirement and Gate Count for the computing unit are shown in Figure 2 and 3 respectively.

FIGURE 2. ROM LUT Estimation

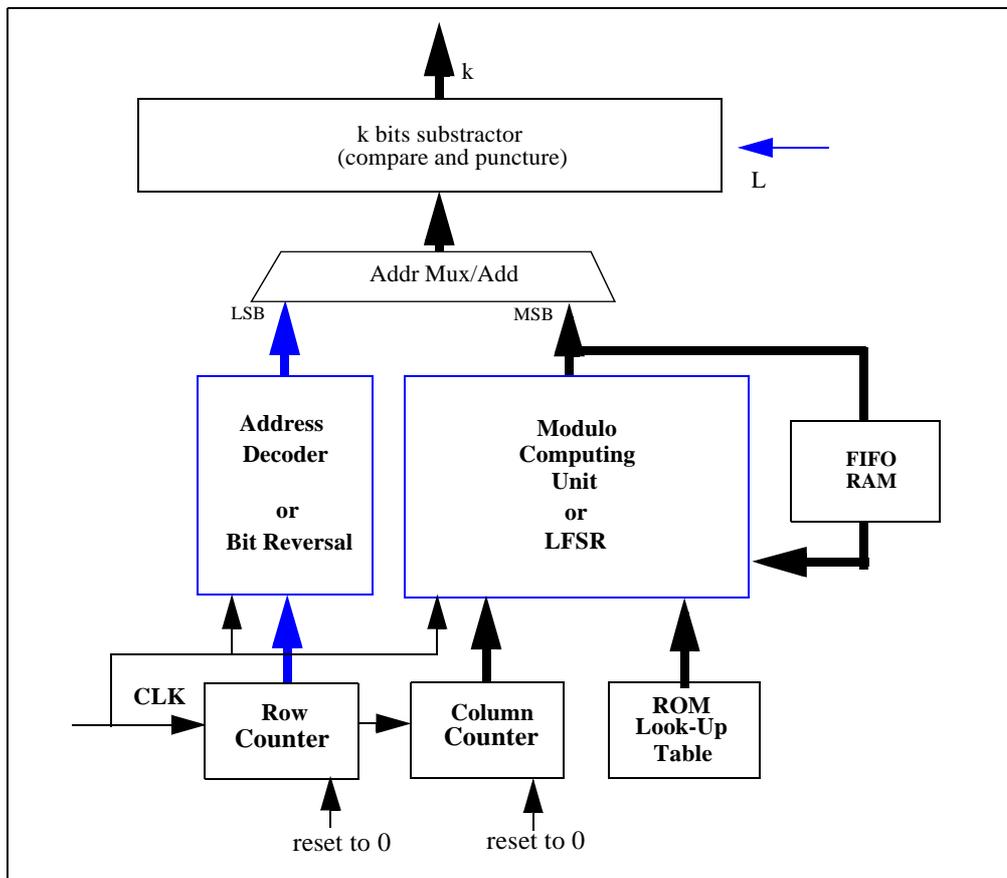


Comments:

The following three interleaver require RAM for the recursive permutation computing.

- Prime Interleaver: $20 \lfloor \log_2 409 \rfloor$ for 20 rows recursive intra-row permutation.
- Prunable Interleaver $32 \lfloor \log_2 316 \rfloor$ for 32 rows recursive intra-row permutation
- Canon Interleaver $\lfloor \log_2 8192 \rfloor$

FIGURE 1. Architecture of Proposed 3GPP Interleaver



0.2 Implementation Complexity Estimation and Comparison for 5 Turbo Interleavers

In what follows, we present a complexity analysis for the ASIC implementation of 5 Turbo Interleavers

0.2.1 ROM Table Requirement

We use the following criterion to calculate the memory requirement for ROM table:

$$\sum_{l=1}^N \lfloor \log_2 c_j \rfloor$$

The ROM table size for Prime/Prunable/Canon/Motorola/OCPNI Interleavers are listed in Table-1

TABLE 3. ROM Table Size for 5 Turbo Interleavers

Prime	Prunable	Canon AL-C	Motorola	OCPNI
1,259bits	2,115bits	12,584bits	50bits	690,228bits

Comments:

- Prime Interleaver parameter table
- Prunable Interleaver parameter table
- Canon AL-C Interleaver requires a single parameter for every 7 bit incremental of frame size
- Motorola Interleaver requires to store LFSR polynomials and partition of row and column dimension parameters (m,n)
- OCPNI requires initial seeds for LFSR and initial seeds for LFSR muxing

0.2.2 Gate Counts for Interleaving Address Computing Units

The common computing platform for the Prime/Prunable/Canon/OCPNI Interleaver is shown in Figure 1. It consists of three major parts (1) Computing unit (2) RAM (FIFO) and (3) Glue logics.

The RAM requirement is listed in Table.5

TABLE 4. RAM Requirement

Prime	Prunable	Canon AL-C	Motorola	OCPNI
2x180bits	2x288bits	2x13bits	NA	NA
1080 gate	1728 gates	78 gates		

TABLE 2. PIP Table for Inter-Row Permutation

r	PIP Π_r	PIP Memory
A	{19,9,14,4,0,2,5,7,12,18,10,8,13,17,3,1,16,6,15,11}	166bits
B	{19,9,14,4,0,2,5,7,12,18,16,13,17,15,3,1,6,11,8,10}	21Bytes
C	{9,8,7,6,5,4,3,2,1,0}	

Once the interleaver optimization is done, the simple implementation of the PIP table is to use the 5 bit decoder based on Boolean logic.

range of the consecutive primes from 17 to 409 (total 73), most of first 19 prime in $\tilde{p}(l)$ table satisfy the co-prime condition and at most 2 prime can not satisfy such a condition. We specify only the position j^* of such an invalid prime in table $\tilde{p}(l)$, which can not be used the cyclic shift for a particular row. For example, for $P_L = 67$, $\tilde{p}(2) = 11$ is not co-prime with 66, therefore the position 2 in table $\tilde{p}(l)$, can not be used for the 3rd row intra-row permutation. See the third column associate with prime 67 is 2. the entry 0 in table 1 indicates the first 19 primes are all valid for the intra-row permutation.

The parameters stored in ROM for the intra-row permutatuion are listed in Table 1.

TABLE 1. Intra-Row Permutation Parameter Table (ROM)

prime	primitive root	Delete position	prime	primitive root	Puncture	prime	primitive root	Puncture	prime	primitive root	Puncture
$N_c^{(l)}$	α_p	l^*	$N_c^{(l)}$	α_p	l^*	$N_c^{(l)}$	α_p	l^*	$N_c^{(l)}$	α_p	l^*
17	3	0	101	2	0	197	2	1	311	17	8
19	2	0	103	5	4	199	3	2	313	10	3
23	5	2	107	2	13	211	2	1	317	2	19
29	2	1	109	6	0	223	3	9	331	3	2
31	3	0	113	3	1	227	2	0	337	10	1
37	2	0	127	3	1	229	6	5	347	2	0
41	6	0	131	2	3	233	3	7	349	2	7
43	3	1	137	3	4	239	7	1/4	353	3	2
47	5	6	139	2	6	241	7	0	359	7	0
53	2	3	149	2	9	251	6	0	367	6	15
59	2	7	151	6	0	257	3	0	373	2	8
61	2	0	157	5	3	263	5	0	379	2	1
67	2	2	163	2	0	269	2	16	384	5	0
71	7	1	167	5	20	271	6	0	389	2	0
73	5	0	173	2	11	277	5	6	397	5	2
79	3	3	179	2	0	281	3	1	401	3	0
83	2	10	181	2	0	283	3	12	409	21	4
89	3	2	191	19	5	293	2	18			
97	5	0	193	5	5	307	5	4			

In addition we have so-called PIP table for inter-row permutation:

0.1 Prime Interleaver Operation

As described in TSGR1#3(99)-217, the prime interleaver performs a stage 2-dimensional matrix permutation. The interleaving address generation is implemented by intra-row permutation and inter-row permutation. The intra-row permutation employs exponential congruent system with a pre-determined parameters. The interleavers optimization is done by searching best parameters for the inter-row permutation.

0.1.1 The Permutation Rules

The intra row permutation row described in TSGR1#4(99)-XXX can be summarized as follows:

- The first row is permuted using recursive computing based on the primitive root α_p of prime P_L :

$$c_1(i) = [\alpha_p \times c_1(i-1)] \text{mod} P_L$$

- The rest rows are constructed by cyclic shift of the first row permutation with a set of primes $p(j)$, i.e.

$$c_j(i) = c_1([i \times p(j)] \text{mod} P_L)$$

- Equivalently, it is a decimated sampling (with rate sampling ratio $p(l)$) of the complete residual system generated by the primitive root α_p .

$$c_j(i) = [\alpha_p^{p(j)} c_j(i-1)] \text{mod} P_L \quad j = 2, 3, \dots R \quad i = 1, 2, \dots C$$

The same intra-row permutation can be applied to the column size with P_L and $P_L + 1$ by padding 0 and P_L at the end of each row.

- At final stage, the inter-row permutation is carried out based on a optimized PIP table.

$$\Pi_r \rightarrow \{1, 2, \dots R\} \rightarrow \{r(1), r(2), \dots r(R)\}$$

For different rows the cyclic shift is designed to use the following set of primes:

$$\tilde{p}(l) = \{7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89\}$$

for row number $l = 2, 3 \dots R$ provided that co-prime condition $\text{gcd}\{\tilde{p}(l), P_L - 1\} = 1$ is satisfied. Since for each column associated with a prime P_L , we need specify such a set of shift offset $p(i)$. In order to reduce the memory requirement, we observed that the, with

TSG-RAN WG-1
Meeting # 4
Yokohama, Japan
Apr. 18-20, 1999

TSG R1#4(99)513

Title: Prime Interleaver Complexity Analysis

Source: Nortel Networks, NTTDoCoMo

0.0 Summary

In this contribution, we present an implementation analysis of Prime Interleaver for 3GPP. The comparison study is also conducted for 5 proposed Turbo Interleavers.