

**Title: ReMCA: A Reconfigurable Multi-Core
Architecture for Full RNS Variant of BFV
Homomorphic Evaluation
(TCAS I)**

Preliminaries

The Textbook BFV Scheme

- $\text{BFV.KeyGen}(\lambda, \omega)$
- $\text{BFV.Enc}(m, pk)$
- $\text{BFV.Dec}(ct, sk)$
- $\text{BFV.HomAdd}(ct_0, ct_1)$
- $\text{BFV.HomMult}(ct_0, ct_1, rlk)$

Preliminaries

Parameter Setup

- polynomial degree: 4096
- the standard deviation of the Gaussian distribution to σ : 3.19
- the size of modulus q : 128 bit(product of four 32 bit primes)?
- the size of the larger modulus Q to at least 288-bit(product of nine 32 bit primes)
- 32-bit primes to construct the RNS for our implementation

Algorithm And Approach

Unified Low-Complexity NTT/INTT

It is a algorithm can control the butterfly unit to do NTT or INTT, decrease the complexity of memory access.

Algorithm 1 Unified Low-Complexity CG NTT/INTT

Let coefficient vector $\mathbf{a} = (a_0, \dots, a_{N-1})$ and $\mathbf{A} = (A_0, \dots, A_{N-1})$. Let ω_N to be the primitive N -th root of unity and $\psi_{2N} = \omega_N^{1/2} \bmod q$.

Input: \mathbf{a} , N , q , sel , ψ_{2N}^i , ψ_{2N}^{-i} , where $i = 1, 2, \dots, N-1$.

Output: $\mathbf{A} = NTT_{\psi_{2N}}^N(\mathbf{a})$ or $INTT_{\psi_{2N}^{-1}}^N(\mathbf{a})$

```
1:  $\mathbf{a} \leftarrow \text{BitReverse}(\mathbf{a})$ 
2: for ( $s = 1; s \leq \log_2 N; s = s + 1$ ) do
3:   for ( $j = 0; j < N/2; j = j + 1$ ) do
4:     if  $sel = 1$  then //NTT
5:        $k_1 = \lfloor j/2^{\log_2(N)-s} \rfloor \cdot 2^{\log_2(N)-s}$ 
6:        $k_2 = N/2^s$ 
7:        $A[j] \leftarrow a[2j] + a[2j + 1] \cdot \psi_{2N}^{2k_1+k_2} \bmod q$ 
8:        $A[j + N/2] \leftarrow a[2j] - a[2j + 1] \cdot \psi_{2N}^{2k_1+k_2} \bmod q$ 
9:     else //INTT
10:       $k_1 = \lfloor \text{BitReverse}(j)/2^{s-1} \rfloor \cdot 2^{s-1}$ 
11:       $k_2 = 2^{s-1}$ 
12:       $A[j] \leftarrow (a[2j] + a[2j + 1]) \cdot (1/2) \bmod q$ 
13:       $A[j + N/2] \leftarrow (a[2j] - a[2j + 1]) \cdot (\psi_{2N}^{-(2k_1+k_2)}/2) \bmod q$ 
14:    end if
15:  end for
16: if  $s \neq \log_2 N$  then
17:    $\mathbf{a} = \mathbf{A}$ 
18: end if
19: end for
20: Return ( $\mathbf{A}$ )
```

Algorithm And Approach

RNS Basis Extension

Algorithm 2 RNS Basis Extension

Let $\mathbf{A}_i \in R_{q_i}$, $q_i^* = q/q_i$, $\tilde{q}_i = (q_i^*)^{-1} \bmod q_i$, $q = \prod_{i=1}^k q_i$,
 $p = \prod_{j=k+1}^{k+k'} q_j$, $Q = q \cdot p$, where $i = 1, \dots, k$, $j = k+1, \dots, k+k'$.

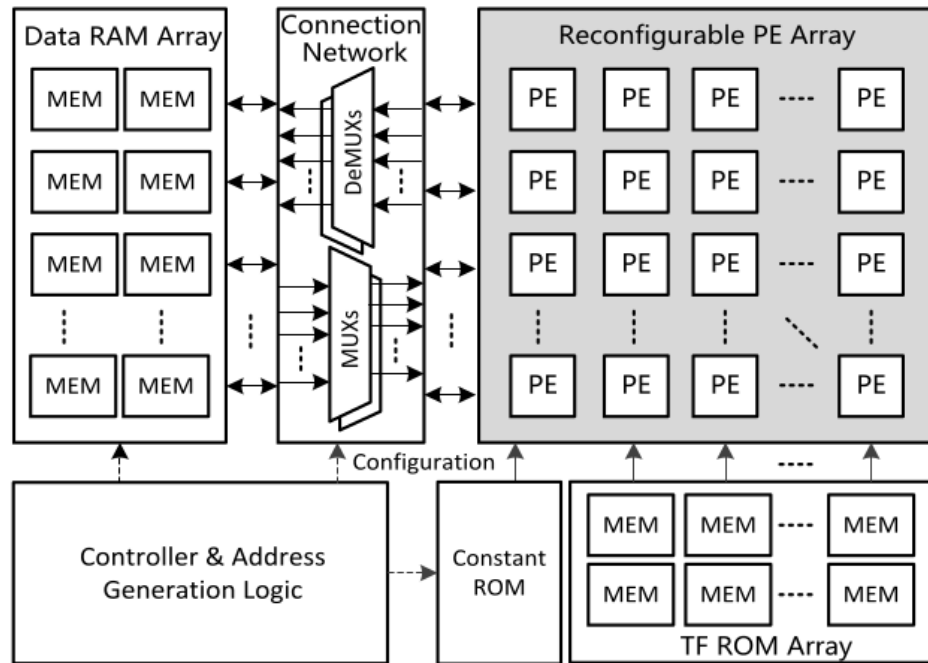
Input: $\mathbf{A}_i, q_i, 1/q_i, \tilde{q}_i, q_i^*, q_j, q$.

Output: $\mathbf{A}_j = \text{BasisExtension}(\mathbf{A}_i)$.

```
1: for ( $i = 1, i \leq k, i = i + 1$ ) do
2:    $\mathbf{A}'_i = \mathbf{A}_i \cdot \tilde{q}_i \bmod q_i$  // Step 1
3: end for
4: for ( $j = k + 1, j \leq k + k', j = j + 1$ ) do
5:    $\mathbf{A}'_j = \sum_{i=1}^k \mathbf{A}'_i \cdot q_i^* \bmod q_j$  // Step 2
6:    $\mathbf{V}_j = \lfloor \sum_{i=1}^k \mathbf{A}'_i \cdot (1/q_i) \rfloor \bmod q_j$  // Step 3
7:    $\mathbf{V}'_j = \mathbf{V}_j \cdot q \bmod q_j$  // Step 4
8:    $\mathbf{A}_j = \mathbf{A}'_j - \mathbf{V}'_j \bmod q_j$  // Step 5
9: end for
10: Return ( $\mathbf{A}_j$ )
```

Architecture

Overall Architecture



Reconfigurable PE Array: perform NTT/INTT, the modular multiplication. One row or multiple rows of PE can be configured as a channel to perform the polynomial arithmetic operations on one RNS base.

TF ROM Array: store the twiddle factor array

Data RAM Array: store the input polynomials, intermediate results and final results

Total of 40 PEs, in which each row corresponds to one channel and each channel contains two slices and 8 PEs (4 for each slice). To maximize the parallelism of the processing path and match the number of extended RNS bases, set five channels in PE array.

Architecture

Reconfigurable PE Unit

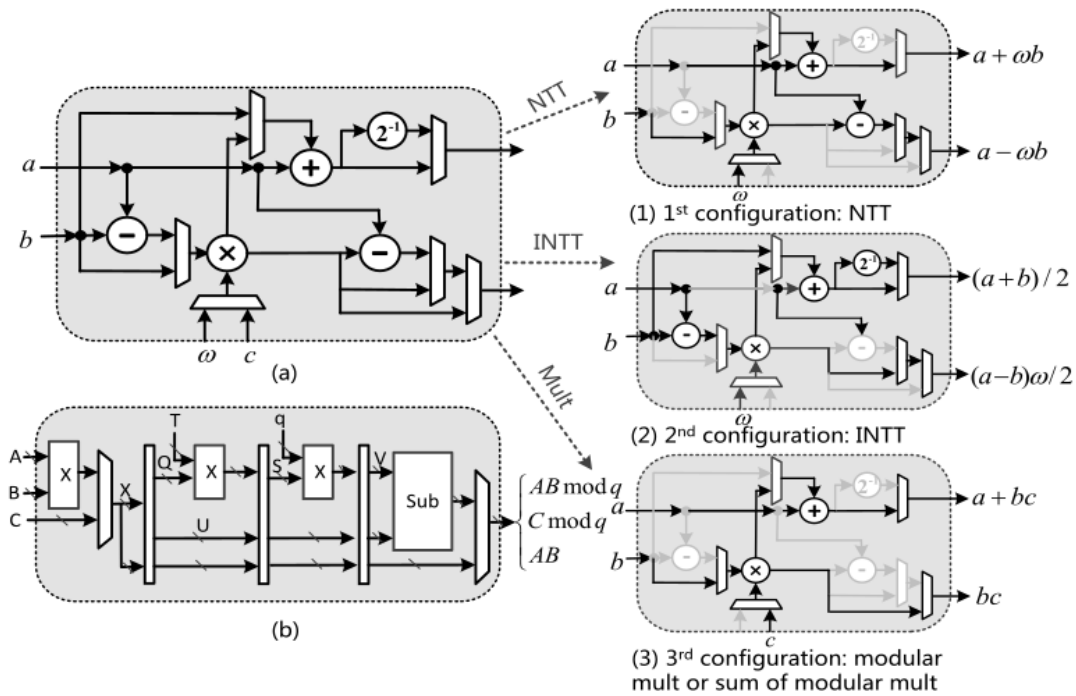


Fig. 2. Architecture of reconfigurable PE. (a) Reconfigurable PE, (b) Barrett modular multiplier.

Reconfigurable PE: INTT,NTT,MULT

Barrett modular multiplier: modular multiplier, modular reduction

1. PE not only supports the functions with the variable modulus, but also supports the summation of modular multiplication

2. By merging the multiplicative factor $1/2$ into the twiddle factors, the reconfigurable PE eliminates the multiplication of $1/2$ in the subtraction path and improves the performance of PE unit.

$$\frac{x}{2} = (2\lfloor \frac{x}{2} \rfloor + 1) \frac{q+1}{2} = \lfloor \frac{x}{2} \rfloor (q+1) + \frac{q+1}{2} = \lfloor \frac{x}{2} \rfloor + \frac{q+1}{2} \pmod{q}$$

3. The Barrett modular multiplier we presented employs a reconfigurable architecture and avoids the needs of other computing units for ReMCA.

Architecture

Conflict-Free Memory Access for NTT/INTT

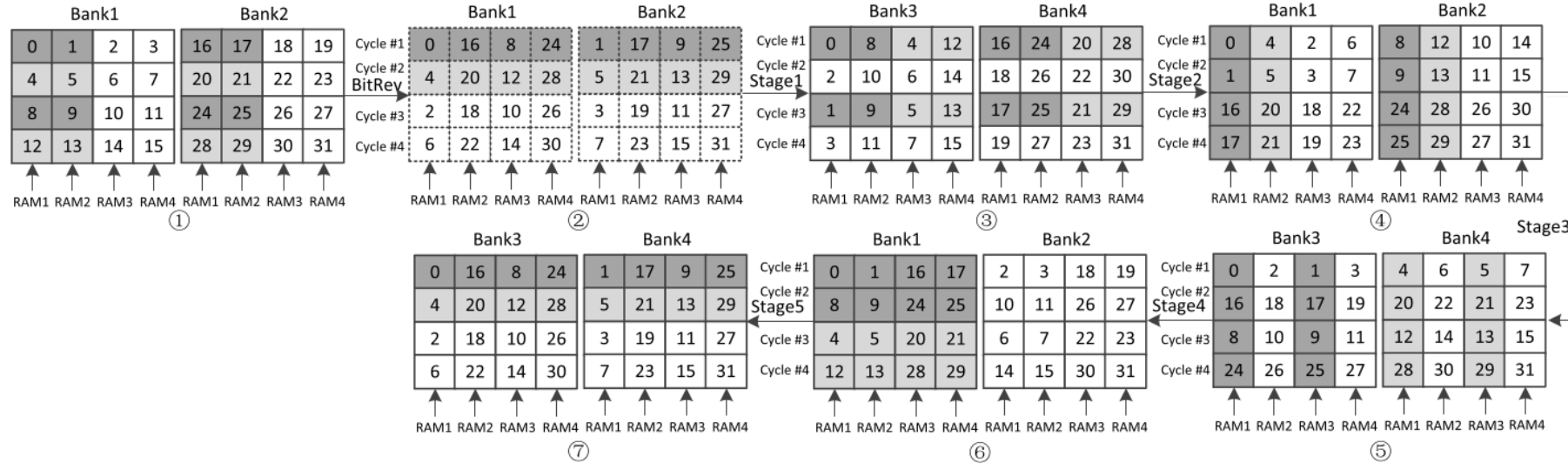


Fig. 3. Data memory access pattern of NTT for $N = 32$, $P = 4$.

Bank is dual-port pattern(could select the bank read port based on PEs)

For the bit-reversal operation, could change the address mapping pattern to avoid timing-consuming or memory-consuming.

Architecture

Unified Computing Model

Unified Hardware Architecture Mapping Model:

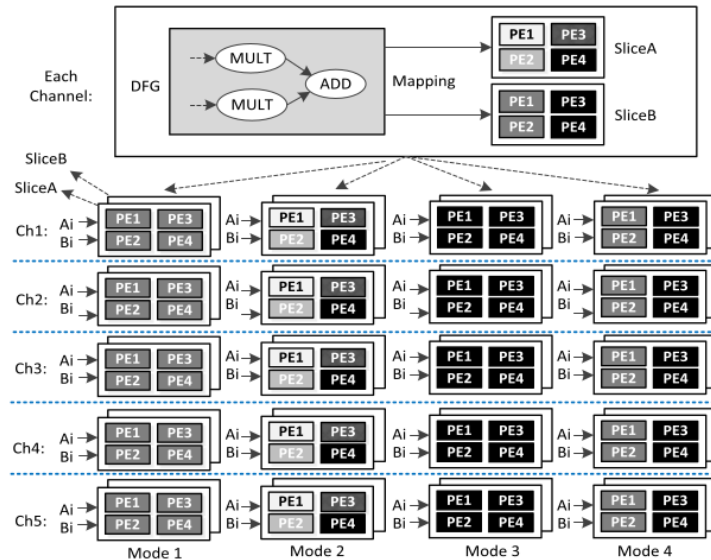


Fig. 4. Unified hardware architecture mapping model of ReMCA.

model 1: compute 32 bits modular mult of four contiguous integers in vector A_i nad vector B_i or four constants in parallel.

model 2: compute the summation of four products, while the inputs of four products are from four different vectors and four constants respectively

model 3: The NTT/INTT transforms are computed using Mode 3

model 4: compute the summation of four products followed by a rounding operation

Architecture

Unified Computing Model

Unified Data Memory Organization Model:

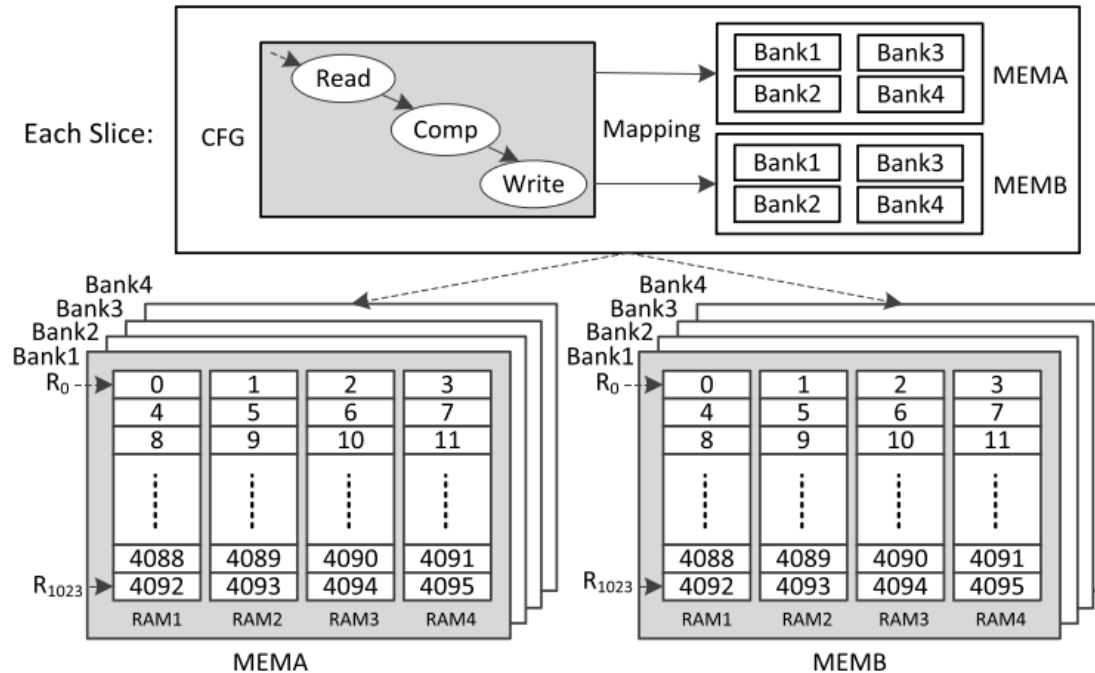


Fig. 5. Unified data memory organization model of ReMCA.

MEM consists of four memory banks, where each memory bank further contains four 1024-depth and 32-bit-width dual-port RAMs. MEMA is used to store the inputs/outputs and intermediates results of almost all functional units in homomorphic evaluation of RNS-BFV except for the NTT and INTT. MEMB is mainly used to store the inputs/outputs and intermediate results of NTT and INTT

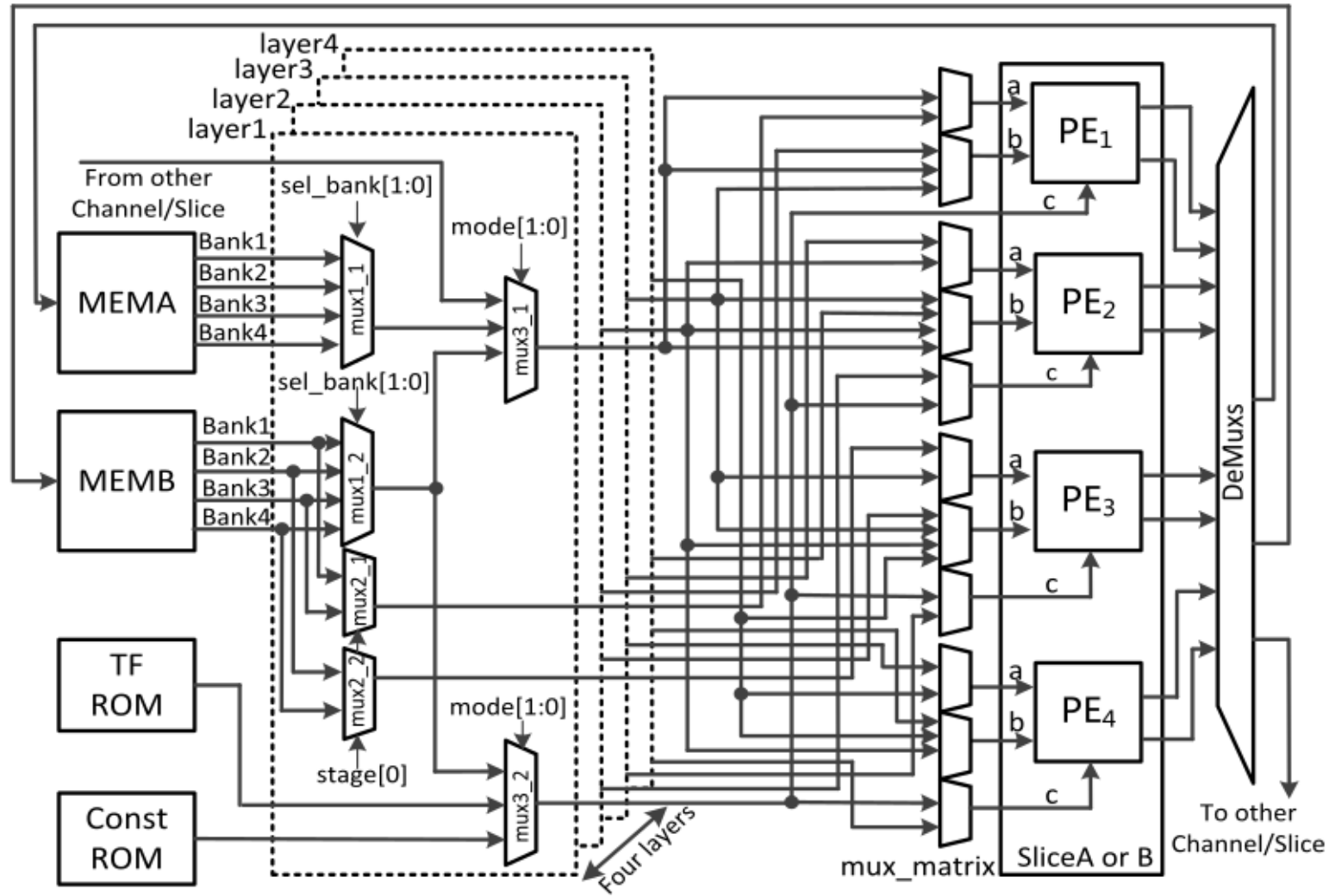


Fig. 6. Detailed structure of interconnection network of each slice.

Mapping Method And Execution Flow

The homomorphic multiplication of RNS-BFV includes four computing units: basis extension, ciphertext multiplication, basis scaling and relinearization

Mapping Method And Execution Flow

Computing Units Mapping

Basis Extension Unit

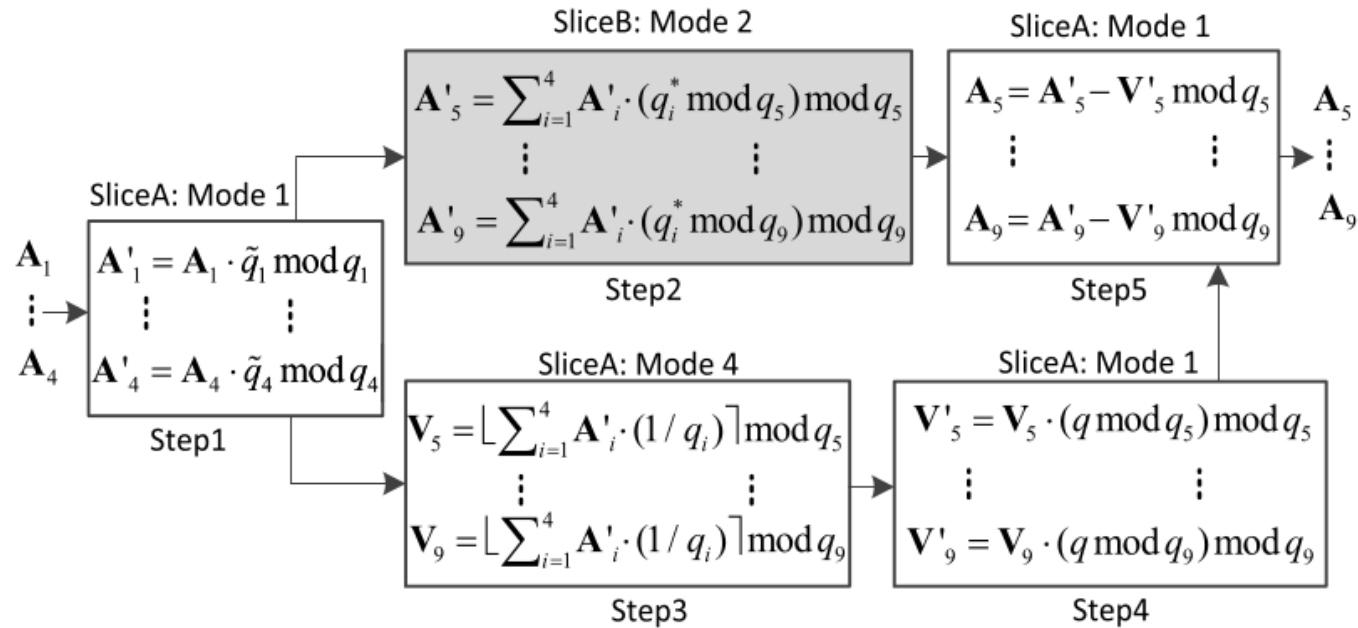


Fig. 7. Mapping method of basis extension unit.

Mapping Method And Execution Flow

Computing Units Mapping

Ciphertext Multiplication Unit

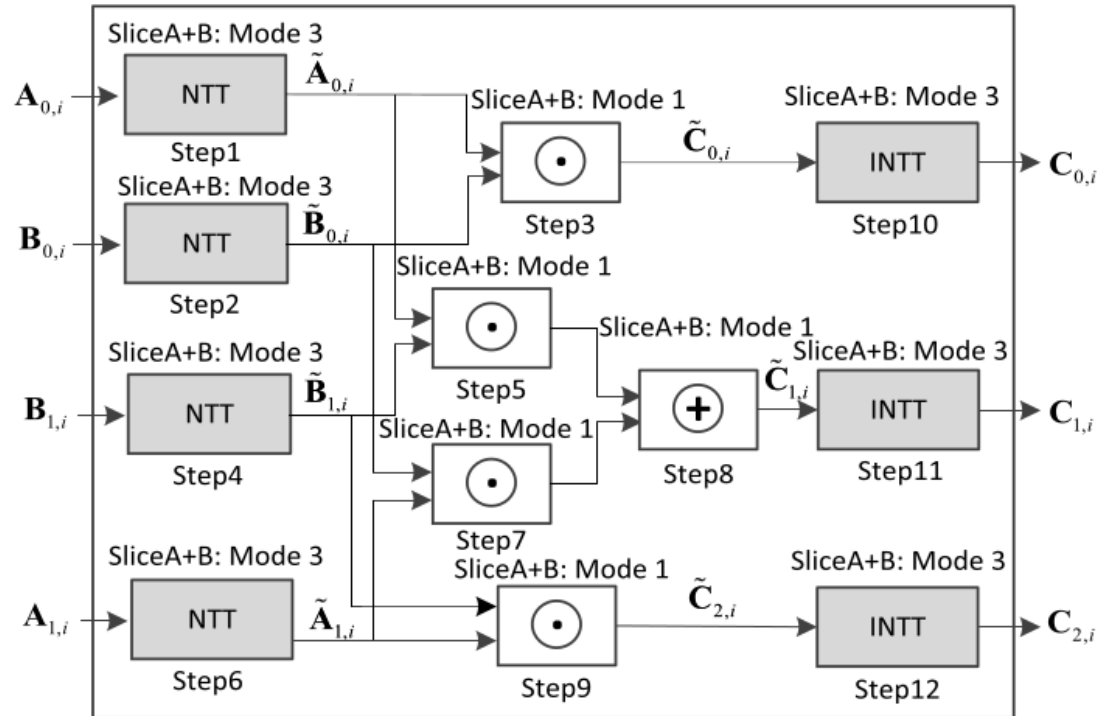


Fig. 8. Mapping method of ciphertext multiplication unit.

Mapping Method And Execution Flow

Computing Units Mapping

Basis Scaling Unit:

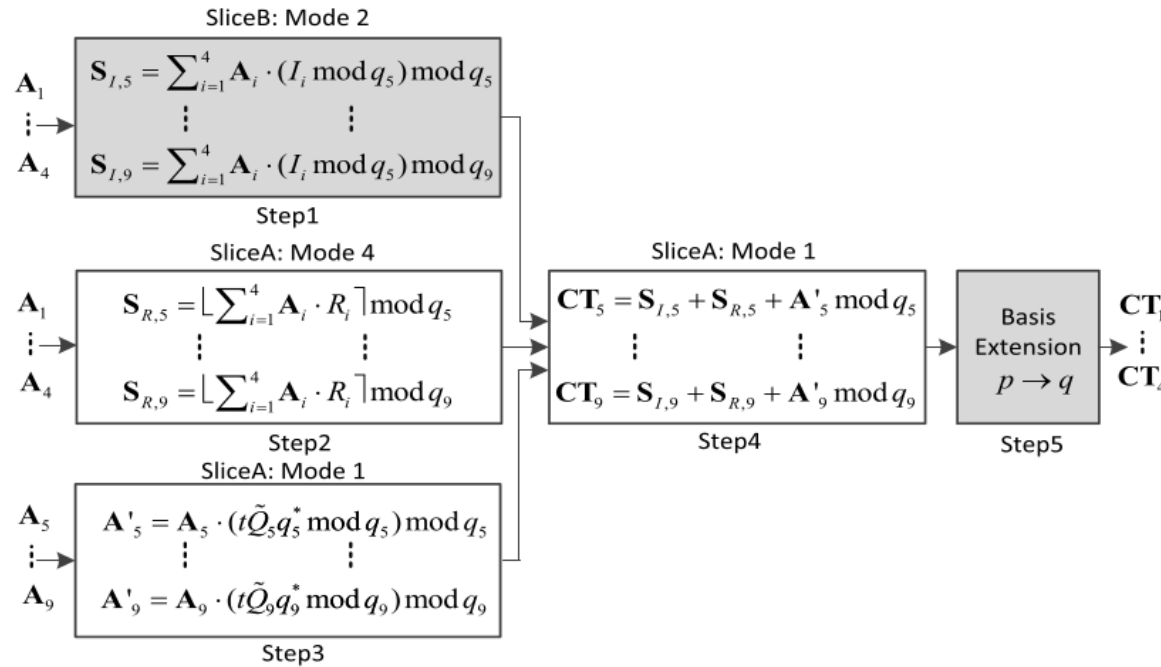


Fig. 9. Mapping method of basis scaling unit.

Execution Flow of RNS-BFV

TABLE III
PERFORMANCE COMPARISON OF NTT/INTT

Design	Platform	N	$\log_2 q$ (bit)	No. bases	Freq (MHz)	Cycles	Time (μ s)	Thr [†] (MB/s)	LUT /ATP ^{††}	FF /ATP ^{††}	BRAM /ATP ^{††}	DSP /ATP ^{††}
Mert [15]	Virtex-7	4096	30	1	200	—	2.3	6793.5	70000 /0.16	— /—	129 /0.30	559 /1.38
ReMCA	Virtex-7	4096	32	1	250	6144	24.58	635.68	5968 /0.15	4394 /0.11	40 /0.98	40 /0.98
Cathébras [14]	Virtex-7	4096	30	4	200	—	150	390.63	41964 /6.30	50961 /7.64	147 /22.05	363 /54.45
ReMCA	Virtex-7	4096	32	4	250	6144	24.58	2542.72	46591 /1.15	35551 /0.87	392 /9.74	400 /9.83
Öztürk [32]	Virtex-7	32768	32	41	250	—	2086.9	2551.02	219192 /457.43	90789 /189.47	193 /402.77	768 /1602.74
ReMCA	Virtex-7	32768	32	41	250	61440	245.76	20853.68	194084 /47.70	153050 /37.61	1759 /432.29	1680 /412.88

[†]Throughput (Thr) = $N \times (\text{No. of bits } (q)) / \text{Time}$. ^{††}ATP=No. LUT or FF or BRAM or DSP multiply the total time (second (column #11, #12) or millisecond (column #13, #14));

IMPLEMENTATION RESULTS AND COMPARISONS

FPGA Implementation Result

Xilinx Vivado tool

Virtex-7 XC7VX1140T

synthesized the design and achieved 250MHz

frequency under the parameter set ($N = 4096$, $\log(q) = 128\text{-bit}$, $\log(q_i) = 32\text{-bit}$)

TABLE I

FPGA IMPLEMENTATION RESULT OF REMCA ON XC7VX1140T

$N=4096$	Slice LUTs	Slice Registers	RAMB36E1	DSP48E1
Total available	712000	1424000	1880	3360
ReMCA	46591 (6.54%)	35551 (2.50%)	392 (20.85%)	400 (11.90%)
Each PE	1043 (0.15%)	811 (0.06%)	—	10 (0.30%)
Each Barrett	560 (0.08%)	553 (0.04%)	—	10 (0.30%)

TABLE II

TIMING RESULTS OF PRIMITIVE OPERATIONS

Operation	Speed	
	(cycles)	(μ s)
Basis Extension	16384	65.54
Ciphertext Mult	47104	188.42
Basis Scaling	24576	98.30
Relinearization	22528	90.11
Total	110592	442.37
Each NTT	6144	24.58
Each INTT	6144	24.58
Each Point-wise Mult	1024	4.10

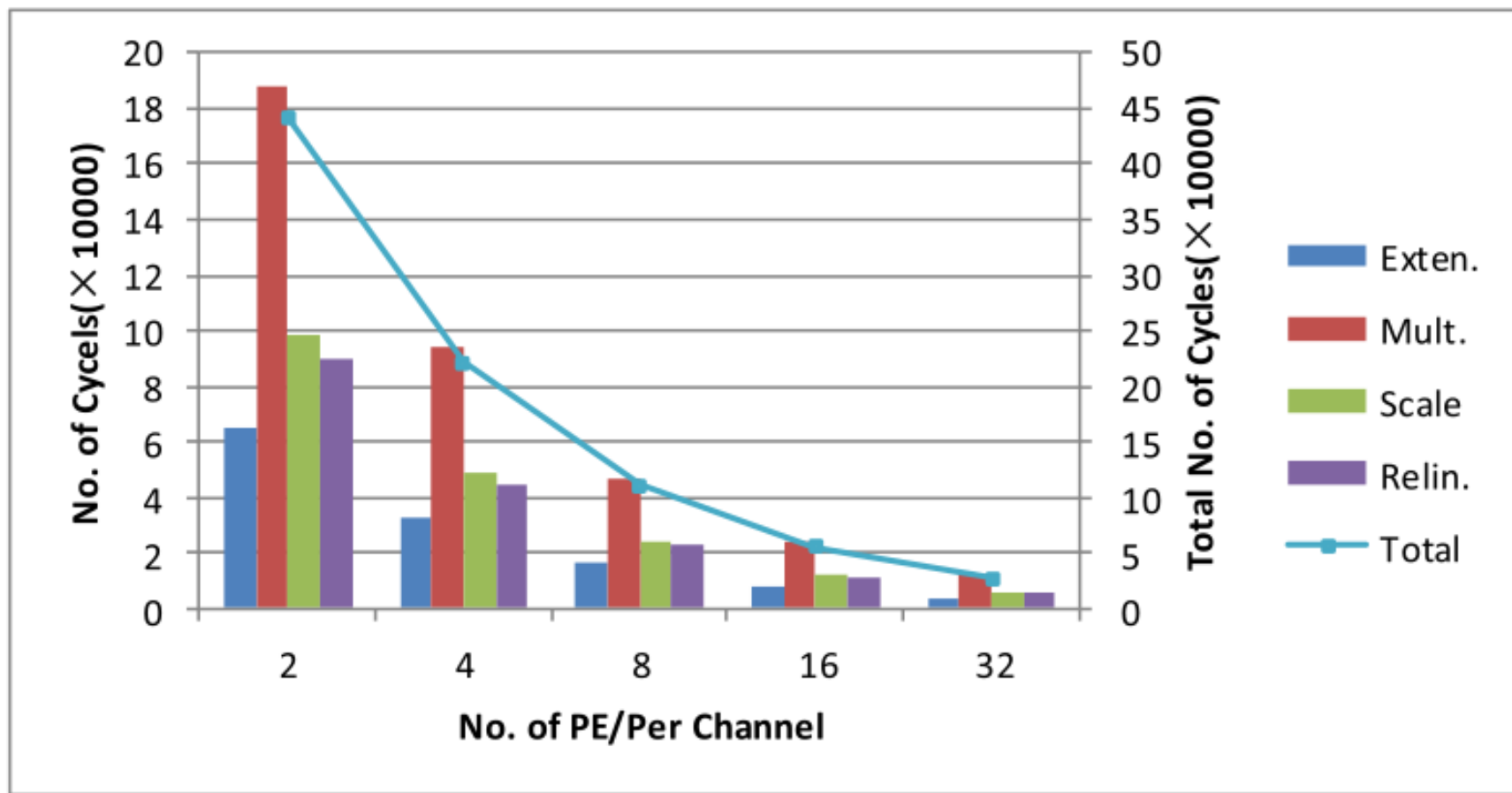


Fig. 12. The relationship between the number of PEs and performance.

Comparison of NTT/INTT Acceleration

TABLE III
PERFORMANCE COMPARISON OF NTT/INTT

Design	Platform	N	$\log q_i$ (bit)	No. bases	Freq (MHz)	Cycles	Time (μ s)	Thr [†] (MB/s)	LUT /ATP ^{††}	FF /ATP ^{††}	BRAM /ATP ^{††}	DSP /ATP ^{††}
Mert [15]	Virtex-7	4096	30	1	200	—	2.3	6793.5	70000 /0.16	— /—	129 /0.30	559 /1.38
ReMCA	Virtex-7	4096	32	1	250	6144	24.58	635.68	5968 /0.15	4394 /0.11	40 /0.98	40 /0.98
Cathébras [14]	Virtex-7	4096	30	4	200	—	150	390.63	41964 /6.30	50961 /7.64	147 /22.05	363 /54.45
ReMCA	Virtex-7	4096	32	4	250	6144	24.58	2542.72	46591 /1.15	35551 /0.87	392 /9.74	400 /9.83
Öztürk [32]	Virtex-7	32768	32	41	250	—	2086.9	2551.02	219192 /457.43	90789 /189.47	193 /402.77	768 /1602.74
ReMCA	Virtex-7	32768	32	41	250	61440	245.76	20853.68	194084 /47.70	153050 /37.61	1759 /432.29	1680 /412.88

[†]Throughput (Thr) = $N \times (\text{No. of bits } (q)) / \text{Time}$. ^{††}ATP=No. LUT or FF or BRAM or DSP multiply the total time (second (column #11, #12) or millisecond (column #13, #14));

Comparison of BFV Acceleration

PERFORMANCE COMPARISON OF HOMOMORPHIC EVALUATION OF RNS-BFV

Design	Platform	N	$\log q_i$ (bit)	No. bases	Freq (MHz)	Cycles	Time (μ s)			Thr (MB/s)	LUT /ATP $\dagger\dagger$	FF /ATP $\dagger\dagger$	BRAM /ATP $\dagger\dagger$	DSP /ATP $\dagger\dagger$
							Mult	Relin	Total					
Cathébras [14]	Virtex-7	4096	30	4	200	—	300	300	600	97.66	54188 /32.51	66444 /39.87	208 /0.12	517 /0.31
ReMCA	Virtex-7	4096	32	4	250	110592	352.3	90.1	442.4	141.27	46591 /20.61	35551 /15.73	392 /0.17	400 /0.18
Roy [17]	Zynq UltraScale+	4096	30	6	200	5349567	—	—	4458	19.72	63522 /283.18	25622 /114.22	388 /1.73	208 /0.93
Turan [18]	Virtex UltraScale+	4096	30	6	200	—	—	—	4340	20.25	57877 /251.19	33989 /147.51	305 $\dagger\dagger\dagger$ /1.32	208 /0.90
ReMCA	Zynq UltraScale+	4096	32	6	380	120832	255.6	58.6	314.2	298.38	64057 /20.13	49307 /15.50	552 /0.17	560 /0.18

\dagger Throughput (Thr) = $N \times (\text{No. of bits } (q)) / \text{Time}$. $\dagger\dagger$ ATP=No. LUT or FF or BRAM or DSP multiply the total time (second); $\dagger\dagger\dagger$ No. BRAM = No. BRAMs (249) + No. URAMs (56).