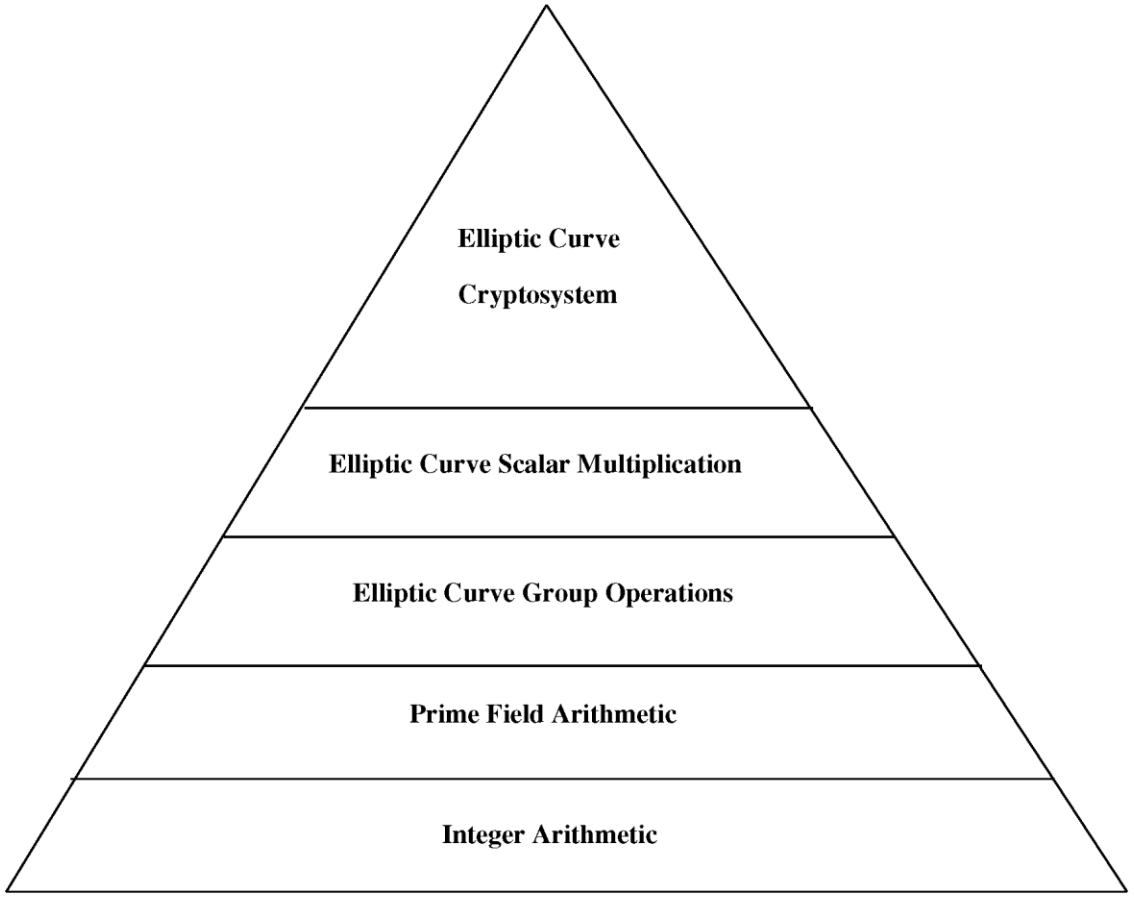
Towards Implementing Carry-Free Primitives for Prime Field

by Vikram Voleti

Promoter: Prof. Dr. Ingrid Verbauwhede
Supervisor: Sujoy Sinha Roy

Contents

1. Introduction



Problem:
Carry Propagation in Integer Arithmetic

2. Our Solution: Carry-Free Arithmetic

"Carry-Free Addition of Recoded Binary Signed-Digit Numbers"
by Behrooz Parhami

- Use of Recoded Binary Signed Digit numbers
- Carry-Free Addition of Recoded Binary Signed Digit numbers
- Use of Carry-Free Addition in other arithmetic operations: subtraction, multiplication, modular reduction

3. Recoded Binary Signed Digit number

RBSD: Numbers represented by {-1, 0, 1}, with no two consecutive 1's or -1's

- n-bit Binary number => 2*(n+1)-bit RBSD number: n+1 bits for sign, n+1 bits for value

3 = [1 1] in binary

= [1 0 -1] in RBSD

3. Recoded Binary Signed Digit number

1. Binary to RBSD Conversion:

A_i	A_{i-1}	X_i
0	0	0
0	1	1
1	0	-1
1	1	0

2. BSD to RBSD Conversion:

A_i	A_{i-1}	A_{i-2}	A_{i-3}	X_i
-1	-1	-1	X	0
		0	-1	0
		0	0	1
		0	1	1
		1	X	1
-1	0	-1	X	1
		0	X	-1
		1	X	-1
-1	1	-1	X	-1
		0	-1	-1
		0	0	0
		0	1	0
		1	X	0
0	-1	-1	X	-1
		0	-1	-1
		0	0	0
		0	1	0
		1	X	0

0	0	X	X	0
0	1	-1	X	0
		0	-1	0
		0	0	1
		0	1	1
		1	X	1
1	-1	-1	X	0
		0	-1	0
		0	0	1
		0	1	1
		1	X	1
1	0	-1	X	1
		0	X	-1
		1	X	-1
1	1	-1	X	-1
		0	-1	-1
		0	0	0
		0	1	0
		1	X	0

These are combinational circuits, there is no delay due to carry propagation.

4. Carry-Free Addition: RBSD numbers

X_i	Y_i	X_{i-1}	Y_{i-1}	S_i
$\bar{1}$	$\bar{1}$	0	0	0
		0	1	0
		1	0	0
		1	1	1
$\bar{1}$	0	0	X	$\bar{1}$
		1	$\bar{1}$	$\bar{1}$
		1	0	$\bar{1}$
		1	1	0
$\bar{1}$	1	0	$\bar{1}$	0
		0	0	0
		1	$\bar{1}$	0
		1	0	0

0	$\bar{1}$	X	0	$\bar{1}$
		$\bar{1}$	1	$\bar{1}$
		0	1	$\bar{1}$
		1	1	0
0	0	$\bar{1}$	$\bar{1}$	$\bar{1}$
		X	0	0
		$\bar{1}$	1	0
		0	X	0
		1	$\bar{1}$	0
		1	1	1
0	1	$\bar{1}$	$\bar{1}$	0
		X	0	1
		0	$\bar{1}$	1
		1	$\bar{1}$	1

1	$\bar{1}$	$\bar{1}$	0	0
		1	1	0
		0	0	0
		0	1	0
1	0	$\bar{1}$	$\bar{1}$	0
		$\bar{1}$	0	1
		$\bar{1}$	1	1
		0	X	1
1	1	$\bar{1}$	$\bar{1}$	$\bar{1}$
		$\bar{1}$	0	0
		0	$\bar{1}$	0
		0	0	0

This is a combinational circuit, there is no delay due to carry propagation.

4. Carry-Free Addition: RBSD numbers

$$S0 = \bar{X}_i^s + Y_i^s + \bar{Y}_i^v$$

$$S1 = \bar{X}_i^v + Y_i^v + \bar{X}_{i-1}^s + \bar{Y}_{i-1}^s$$

$$S2 = X_i^v + \bar{Y}_i^v + \bar{X}_{i-1}^s + \bar{Y}_{i-1}^s$$

$$S3 = \bar{X}_i^s + Y_i^s + \bar{X}_{i-1}^v + Y_{i-1}^s + \bar{Y}_{i-1}^v$$

$$S4 = X_i^v + \bar{Y}_i^s + X_{i-1}^s + \bar{X}_{i-1}^v + \bar{Y}_{i-1}^v$$

$$S_i^s = S0.S1.S2.S3.S4.(\bar{X}_i^v + \bar{Y}_i^s).(X_i^s + Y_i^s + X_{i-1}^s).(X_i^s + Y_i^s + Y_{i-1}^s)$$

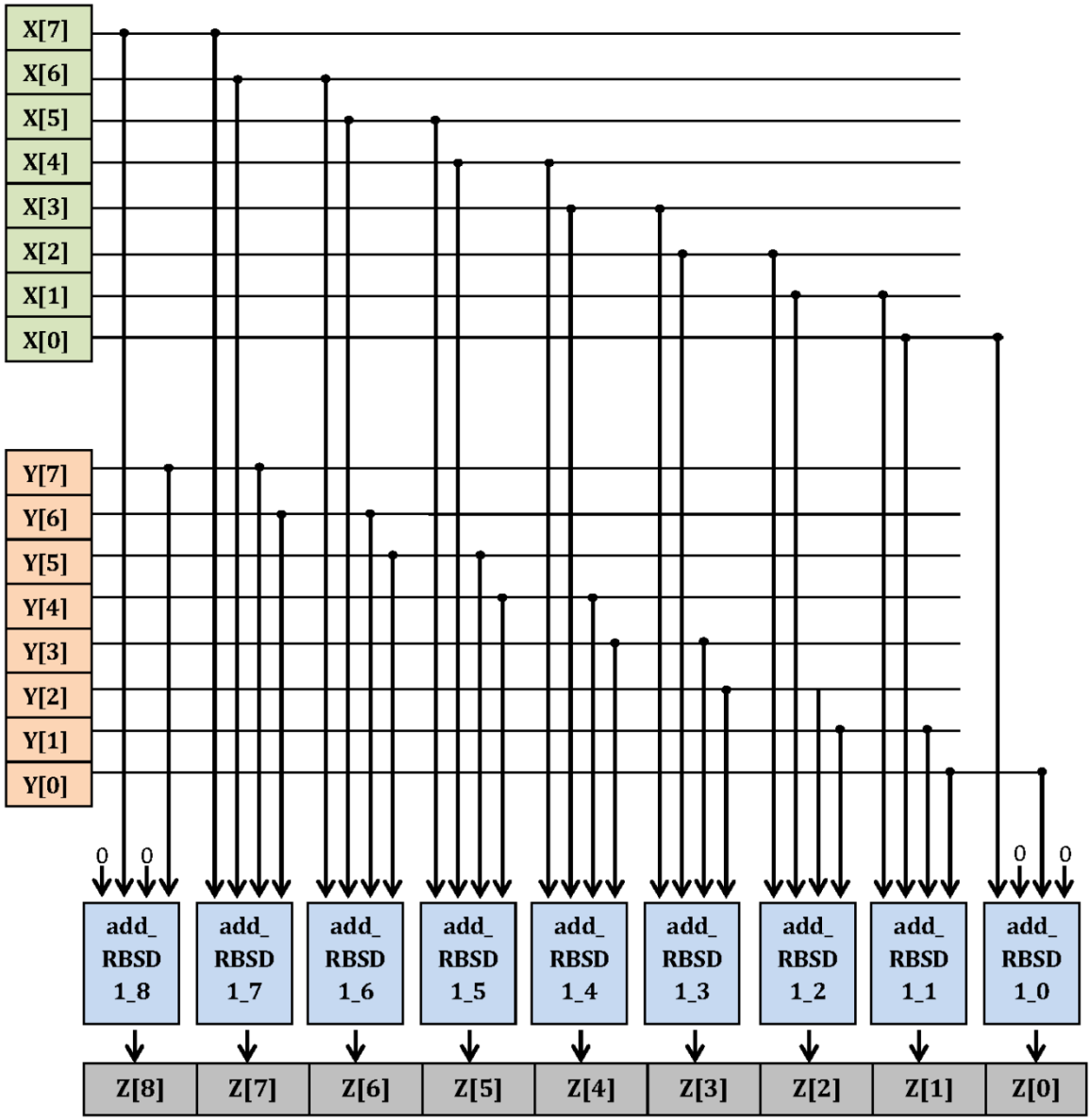
$$S_i^v = S0.S1.S2.S3.S4.(X_i^s + \bar{X}_i^v + \bar{Y}_i^s).(\bar{X}_i^v + \bar{Y}_i^v + X_{i-1}^v).(\bar{X}_i^v + \bar{Y}_i^v + Y_{i-1}^v).$$

$$(X_i^v + Y_i^v + X_{i-1}^v).(X_i^v + Y_i^v + Y_{i-1}^v).(X_i^v + Y_i^v + \bar{X}_{i-1}^s + Y_{i-1}^s).(X_i^v + Y_i^v + X_{i-1}^s + \bar{Y}_{i-1}^s)$$

Maximum combinational delay:
69-bit Carry-Free RBSD Addition: 4.784ns
133-bit Carry-Free RBSD Addition: 4.784ns

For Carry-Free addition of two 132-bit (each of sign and value) RBSD numbers is:
Selected Device : 5vlx30ff324-3 (Virtex-5)
Slice Logic Utilization: Number of Slice LUTs: 928 out of 19200 4%
Slice Logic Distribution: Number of LUT Flip Flop pairs used: 928

4. Carry-Free Addition: RBSD numbers



4. Carry-Free Addition: binary numbers

A_i	A_{i-1}	A_{i-2}	B_i	B_{i-1}	B_{i-2}	S_i
0	0	0	0	0	0	0
			0	0	1	0
			0	1	0	1
			0	1	1	1
			1	0	0	$\bar{1}$
			1	0	1	$\bar{1}$
			1	1	0	0
			1	1	1	0
0	0	1	0	0	0	0
			0	0	1	1
			0	1	0	1
			0	1	1	1
			1	0	0	$\bar{1}$
			1	0	1	0
			1	1	0	0
			1	1	1	0
0	1	0	0	0	0	1
			0	0	1	1
			0	1	0	$\bar{1}$
			0	1	1	0
			1	0	0	0
			1	0	1	0
			1	1	0	0
			1	1	1	1
0	1	1	0	0	0	1
			0	0	1	1
			0	1	0	0
			0	1	1	0
			1	0	0	0
			1	0	1	0
			1	1	0	1
			1	1	1	1

1	0	0	0	0	0	$\bar{1}$
			0	0	1	$\bar{1}$
			0	1	0	0
			0	1	1	0
			1	0	0	0
			1	0	1	0
			1	1	0	$\bar{1}$
			1	1	1	$\bar{1}$
1	0	1	0	0	0	$\bar{1}$
			0	0	1	0
			0	1	0	0
			0	1	1	0
			1	0	0	0
			1	0	1	1
			1	1	0	$\bar{1}$
			1	1	1	$\bar{1}$
1	1	0	0	0	0	0
			0	0	1	0
			0	1	0	0
			0	1	1	1
			1	0	0	$\bar{1}$
			1	0	1	$\bar{1}$
			1	1	0	$\bar{1}$
			1	1	1	0
1	1	1	0	0	0	0
			0	0	1	0
			0	1	0	1
			0	1	1	1
			1	0	0	$\bar{1}$
			1	0	1	$\bar{1}$
			1	1	0	0
			1	1	1	0

This is a combinational circuit, there is no delay due to carry propagation.

4. Carry-Free Addition: binary numbers

$$S0 = A_i + A_{i-1} + \overline{B}_i + \overline{B}_{i-1}$$

$$S1 = A_i + A_{i-1} + \overline{A}_{i-2} + \overline{B}_i + B_{i-1} + \overline{B}_{i-2}$$

$$S2 = A_i + \overline{A}_{i-1} + A_{i-2} + B_i + \overline{B}_{i-1} + \overline{B}_{i-2}$$

$$S3 = \overline{A}_i + A_{i-1} + B_i + \overline{B}_{i-1}$$

$$S4 = \overline{A}_i + A_{i-1} + \overline{A}_{i-2} + B_i + B_{i-1} + \overline{B}_{i-2}$$

$$S5 = \overline{A}_i + \overline{A}_{i-1} + A_{i-2} + \overline{B}_i + \overline{B}_{i-1} + \overline{B}_{i-2}$$

$$S6 = \overline{A}_i + \overline{A}_{i-1} + \overline{A}_{i-2} + \overline{B}_i + \overline{B}_{i-1}$$

$$S_i^s = S0.S1.S2.S3.S4.S5.S6.(A_i + A_{i-1} + B_i).(A_i + \overline{A}_{i-1} + A_{i-2} + B_i + B_{i-1})$$
$$.(A_i + \overline{A}_{i-1} + A_{i-2} + \overline{B}_i).(A_i + \overline{A}_{i-1} + \overline{A}_{i-2}).(\overline{A}_i + A_{i-1} + \overline{B}_i + B_{i-1}).(\overline{A}_i + \overline{A}_{i-1} + B_i)$$

$$S_i^v = S0.S1.S2.S3.S4.S5.S6.(A_i + A_{i-1} + A_{i-2} + B_i + B_{i-1}).(A_i + A_{i-1} + \overline{A}_{i-2} + B_i + B_{i-1})$$
$$.(A_i + \overline{A}_{i-1} + \overline{B}_i + B_{i-1}).(A_i + A_{i-1} + A_{i-2} + \overline{B}_i + \overline{B}_{i-1} + B_{i-2}).(A_i + \overline{A}_{i-1} + \overline{A}_{i-2} + B_i + \overline{B}_{i-1})$$
$$.(\overline{A}_i + A_{i-1} + A_{i-2} + \overline{B}_i + B_{i-1}).(\overline{A}_i + A_{i-1} + \overline{A}_{i-2} + \overline{B}_i + B_{i-1} + B_{i-2}).(\overline{A}_i + \overline{A}_{i-1} + B_i + B_{i-1})$$
$$.(\overline{A}_i + \overline{A}_{i-1} + A_{i-2} + B_i + \overline{B}_{i-1} + B_{i-2})$$

Maximum combinational delay:

Carry_Free_Add_Bin32: 4.028ns

Carry_Free_Add_Binl32: 4.028ns

For Carry-Free addition of two 132 bit binary numbers:

Selected Device : 5vlx30ff324-3 (Virtex-5)

Slice Logic Utilization: Number of Slice LUTs: 265 out of 19200 1%

Slice Logic Distribution: Number of LUT Flip Flop pairs used: 265

5. Carry-Free Subtraction of RBSD numbers

A - B = A + (-B)

3 = [1 1] in binary = [1 0 -1] in RBSD	-3 = [1 0 1] in 2's Complement = [-1 0 1] in RBSD

To negate an RBSD number: invert the 1's and -1's:

```
Xs_new = ~Xs & Xv;
Xv_new = Xv;
```

Adder/Subtractor circuit for 256 bits:

Maximum combinational path delay: 5.402ns

Selected Device : 5vlx30ff324-3
Slice Logic Utilization: Number of Slice LUTs: 1802 out of 19200 9%
Slice Logic Distribution: Number of LUT Flip Flop pairs used: 1802

6. Multiplication of 66-bit binary numbers

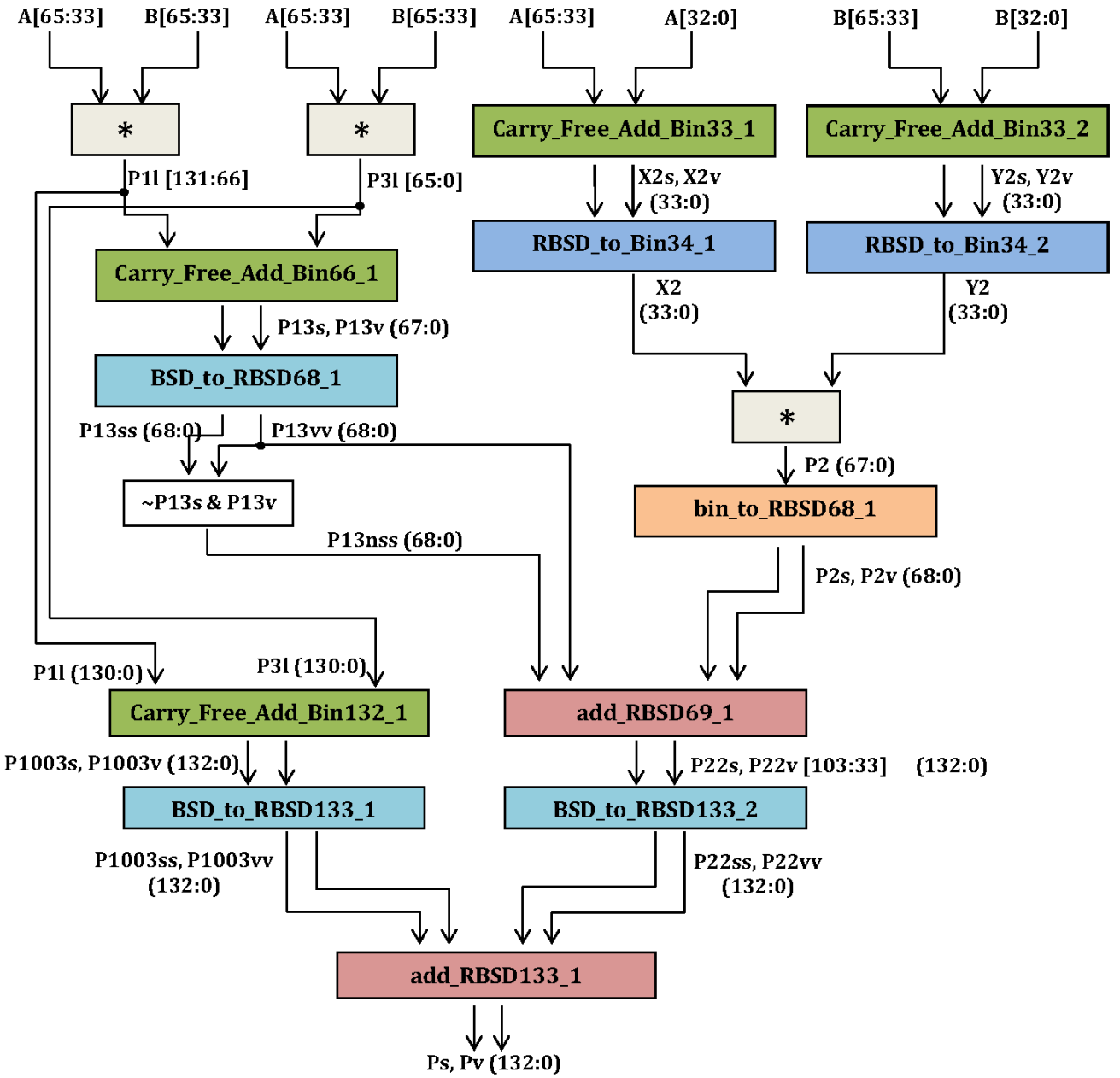
Karatsuba Multiplication Algorithm for 66-bit numbers:

P = ((Ah*Bh)*2^66 + Al*Bl)+ ((Ah+Al)*(Bh+Bl) - (Ah*Bh+Al*Bl))*2^33

Karatsuba Multiplication of 256-bit binary numbers is performed using 66-bit Karatsuba Multiplication incorporating Carry-Free Addition.

6. Multiplication of 66-bit binary numbers

Using Karatsuba Multiplication Algorithm



6. Multiplication of 66-bit binary numbers

Maximum combinational path delays of individual modules:

Carry_Free_Add_Bin32: 4.028ns
Carry_Free_Add_Bin64: 4.028ns
Carry_Free_Add_Bin132: 4.028ns

BSD_to_RBSD68: 4.028ns
BSD_to_RBSD132: 4.028ns

Add_RBSD69: 4.784ns
Add_RBSD133: 4.784ns

RBSD_to_Bin33: 12.824ns

Bin_to_RBSD68: 3.607ns

Maximum combinational path delay of Multiplier: 24.978ns

Slice Logic Utilization: Number of Slice LUTs: 2223 out of 19200 11%
Slice Logic Distribution: Number of LUT Flip Flop pairs used: 2223

7. Modular Reduction

Fast reduction modulo $p256 = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$

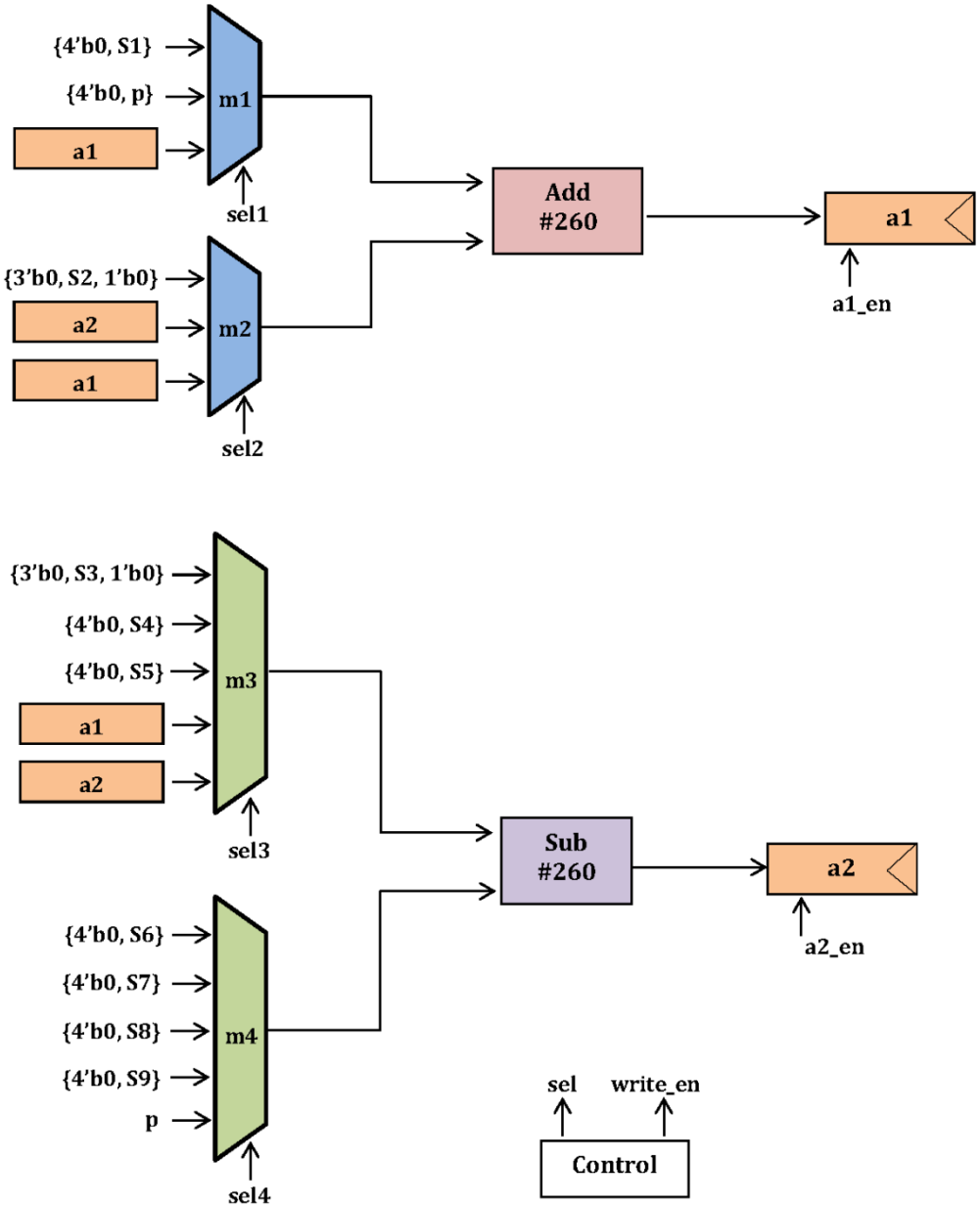
INPUT: An integer $c = (c15, \ldots, c2, c1, c0)$ in base 2^{32} with $0 \le c < p256^2$.

OUTPUT: $c \bmod p256$.

1. Define 256-bit integers:
$s1 = (c7, c6, c5, c4, c3, c2, c1, c0)$,
$s2 = (c15, c14, c13, c12, c11, 0, 0, 0)$,
$s3 = (0, c15, c14, c13, c12, 0, 0, 0)$,
$s4 = (c15, c14, 0, 0, 0, c10, c9, c8)$,
$s5 = (c8, c13, c15, c14, c13, c11, c10, c9)$,
$s6 = (c10, c8, 0, 0, 0, c13, c12, c11)$,
$s7 = (c11, c9, 0, 0, c15, c14, c13, c12)$,
$s8 = (c12, 0, c10, c9, c8, c15, c14, c13)$,
$s9 = (c13, 0, c11, c10, c9, 0, c15, c14)$.

2. Return$(s1 + 2s2 + 2s3 + s4 + s5 - s6 - s7 - s8 - s9 \bmod p256)$.

7. Modular Reduction



7. Modular Reduction

This is a sequential circuit, it 10 clock cycles to complete.

Minimum period: 5.215ns (Maximum Frequency: 191.755MHz)

Minimum input arrival time before clock: 5.880ns

Maximum output required time after clock: 7.229ns

Maximum combinational path delay: 7.895ns

Selected Device : 5vlx30ff324-3 (Virtex-5)
Slice Logic Utilization: Number of Slice Registers: 1051 out of 19200 5%
Number of Slice LUTs: 10612 out of 19200 55%

8. Problems

Binary -> RBSD conversion:

- It is assumed that binary number is positive
- This conversion increases bit size by 1, and then doubles the bit size for inclusion of sign

Carry-Free Addition:

- Results in a BSD number that is not necessarily RBSD; fortunately, BSD to RBSD conversion is combinational

RBSD -> Binary conversion:

- Involves carry propagation

Karatsuba Multiplication with Carry-Free Addition incorporated:

- Splitting a BSD or an RBSD number can result in carry propagation
- Have to use 64-bit RBSD to Binary converter in which Carry Propagation occurs, instead of faster sequential 32-bit RBSD to Binary conversion
- Ultimately, integer multiplication is performed, which involves Carry Propagation

9. Future Work

- Implement 2's complement binary number to RBSD conversion
- Implement carry-free integer multiplication
- Incorporate splitting and concatenation of BSD and RBSD numbers in carry-free arithmetic
- Design of ECC Processor that works with Carry-Free arithmetic:
    - design of arithmetic modules that work completely in the BSD number domain, without conversions to and from binary

References

1. Behrooz Parhami, "Carry-Free Addition of Recoded Binary Signed-Digit Numbers", IEEE Transactions on Computers, Vol. 37, No. 11, November 1988

2. Behrooz Parhami, "Generalized Signed-Digit Number Systems: A Unifying Framework for Redundant Number Representations", IEEE Transactions on Computers, , Vol. 39, No. 1, January 1990

3. Darrel Hankerson, Alfred Menezes, Scott Vanstone, "Guide to Elliptic Curve Cryptography"

*All the modules described have also been coded in Octave to verify the results.

*For long integer calculation, GP/PARI calculator has also been used.

Thank you!