Jeremy Barthélemy

ECE 645

**Montgomery Multiplication**

*Pseudocode: Architecture 2*

Pseudocode below is based on the Multiple-Word Radix-2 Montgomery Multiplication Algorithms found in the Huang IEEE paper.

Note, i=0∑n-1 I signifies the sum from i=0 to i=(n-1) of I.

General Algoithm:

Input:

odd M; n = ⌊ log2M ⌋ + 1; word size w, e = ⌈ (n+1)/w ⌉,

X = i=0∑n-1 xi\*2i, Y = j=0∑e-1 Y(j)\*2w\*j,M = j=0∑e-1 M(j)\*2w\*j with 0 ≤ X,Y < M

Output:

Z = j=0∑e-1 S(j)\*2w\*j = MP(X,Y,M) ≡ X\*Y\*2-n (mod M), 0 ≤ Z < 2M

S = 0; /\*initialize all words of S\*/

**for** i = 0 **to** n-1 **do**

qi = (xi\*Y) ⊕ S0(0);

(C(1),S(0)) = xi\*Y(0) + qi\*M(0) + S(0);

**for** j = 1 **to** e **step** 1 **do**

(C(j+1), S(j)) = C(j) + xi\*Y(j) + qi\*M(j) + S(j);

S(j-1) = (S0(j),S(j-1)w-1..1);

S(e) = 0;

**return** Z = S;

Computation in Task D:

Input:

xi, Y(0), M(0), S0(1), S(0)w-1..1

Output:

qi, C(1), S(0)w-1..1

qi = (xi\*Y0(0)) ⊕ S1(0);

(CO(1), SO(0)w-1, S(0)w-2..0) = (1, S(0)w-1..1) + xi\*Y(0) + qi\*M(0);

(CE(1), SE(0)w-1, S(0)w-2..0) = (0, S(0)w-1..1) + xi\*Y(0) + qi\*M(0);

**if** S0(1) = 1 **then**

C(1) = CO(1);

S(0)w-1..1 = (SO(0)w-1, S(0)w-2..1);

**else**

C(1) = CE(1);

S(0)w-1..1 = (SE(0)w-1, S(0)w-2..1);

Computation in Task E:

Input:

qi, xi, C(j), Y(j), M(j), S0(j+1), S(j)w-1..1

Output:

C(j+1), S(j)w-1..1, S0(j)

(CO(j+1), SO(j)w-1, S(j)w-2..0) = (1, S(j)w-1..1) + C(j) + xi\*Y(j) + qi\*M(j);

(CE(j+1), SE(j)w-1, S(j)w-2..0) = (1, S(j)w-1..1) + C(j) + xi\*Y(j) + qi\*M(j);

**if** S0(j+1) = 1 **then**

C(j+1) = CO(j+1);

S(j)w-1..1 = (SO(j)w-1, S(j)w-2..1);

**else**

C(j+1) = CE(j+1);

S(j)w-1..1 = (SE(j)w-1, S(j)w-2..1);

Computation in Task F:

Input:

qi, xi, C(e-1), Y(e-1 ), M(e-1), S(e-1)w-1..1, C0(e)

Output:

C(e), S(e-1)w-1..1, S0(e-1)

(C(e), S(e-1)) = (C0(e), S(e-1)w-1..1) + C(e-1) + xi\*Y(e-1) + qi\*M(e-1);