SIMD Acceleration of Modular Arithmetic on Contemporary Embedded Platforms

Krishna Pabbuleti **Deepak Mane** **Avinash Desai**

Curt Albert **Patrick Schaumont**

Bradley Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA

Contents

- High Performance Crypto in Embedded Systems
- Modular Multiplication in ECC
- Two Methods of SIMD Modular Multiplication
  - Partial Accumulator Method
  - Multiplicand Reduction Method
- Results
- Conclusion

High Performance Crypto in Embedded Systems

- **Why crypto in Embedded Systems?**
  - **Cornerstone for most security features**
- **Why Public Key Crypto?**
  - **Necessary for advanced security features**
- **PKC is also the most performance hungry**

- **Challenge: How to map efficiently on Embedded Platforms?**

Elliptic Curve Crypto

- **Why ECC?**
  - **Shorter key lengths**
- **What is the computational bottleneck?**
  - **90% of ECC is modular multiplications**
  - **Word lengths ranging from 192 to 512 bits**

Modular Arithmetic

- **Clock arithmetic**
  - **5 mod 13 = 5**
  - **14 mod 13 = 1**
- **Modular Multiplication**
  - **(5 mod 13) * (7 mod 13) = (35 mod 13) = 9 mod 13**

- **Features of ECC Modular Arithmetic**
  - **Typical word lengths 192 to 512 bits**
  - **Primes in ECC are standardized and known beforehand**

NIST Primes

- **NIST primes are the primes of the form $2^n \pm 2^m \pm \ldots\ldots \pm 1$.**
- **The five NIST primes defined are:**
  - **P192 = $2^{192} - 2^{64} - 1$**
  - **P224 = $2^{224} - 2^{96} + 1$**
  - **P256 = $2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$**
  - **P384 = $2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$**
  - **P521 = $2^{521} - 1$**
- **NIST Primes allow efficient reduction**
  - **$2^{192} = 2^{64} + 1 \pmod{P192}$**
  - **And so on**

SIMD Modular Multiplication

- **Accelerate modular multiplication using SIMD**
- **SIMD: Vectorized computation**
  - **Eg: {d1, d0} x {d3, d2} = {d1xd3, d0xd2}**
  - **ARM has NEON and ATOM has SSE2**
  - **Both can do 128-bit vector processing**

- **Two approaches to perform SIMD modular multiplication**
  - **Partial Accumulator Method**
  - **Multiplicand Reduction Method**

Partial Accumulators Method

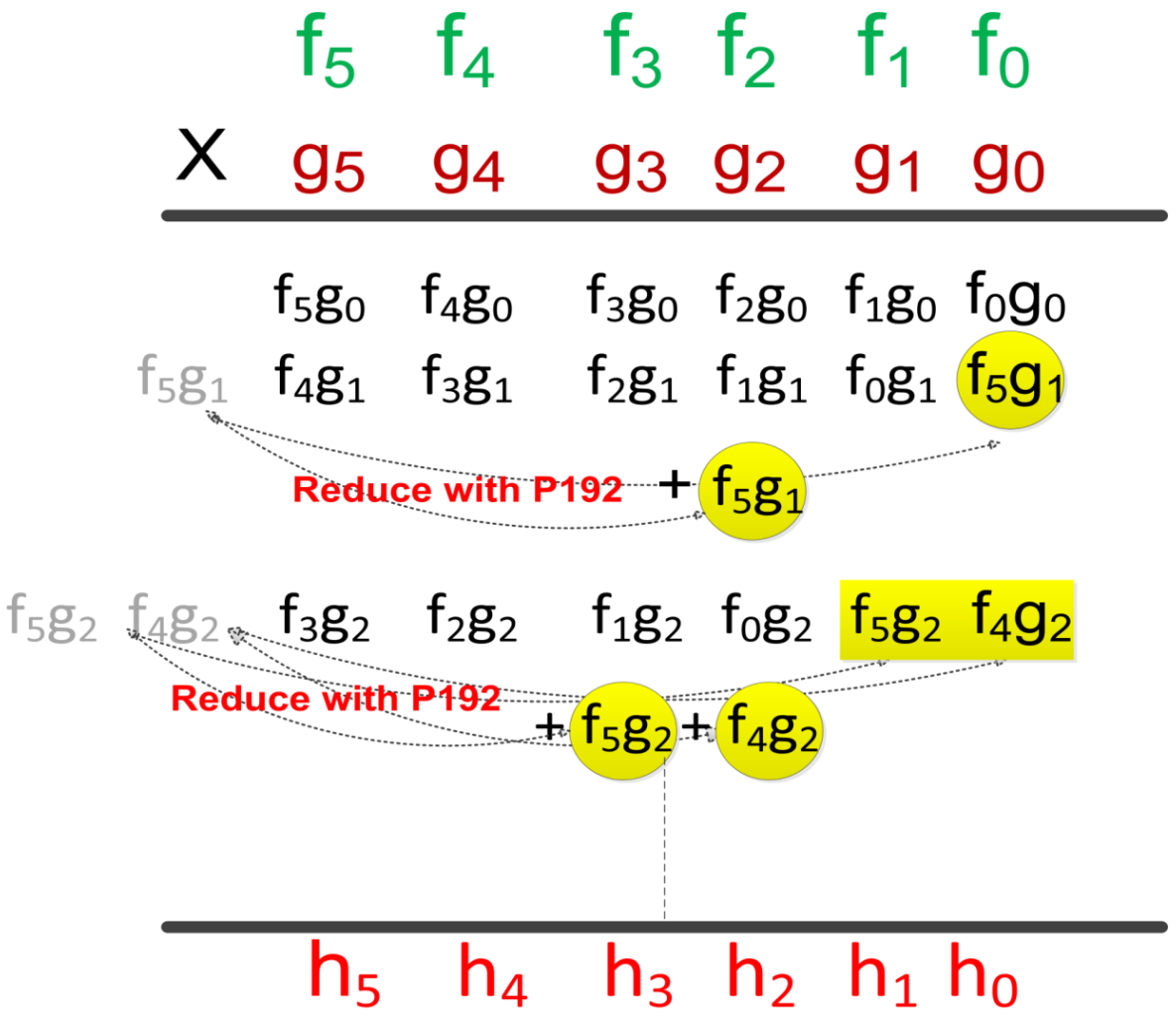
- **$P192 = 2^{192} - 2^{64} - 1$**
- **$2^{192} = (2^{64} + 1) \bmod P192$**
- **A 192-bit number is represented in polynomial form**

  $f = f_0.2^0 + f_1.2^{32} + f_2.2^{64} + f_3.2^{96} + f_4.2^{128} + f_5.2^{160}$

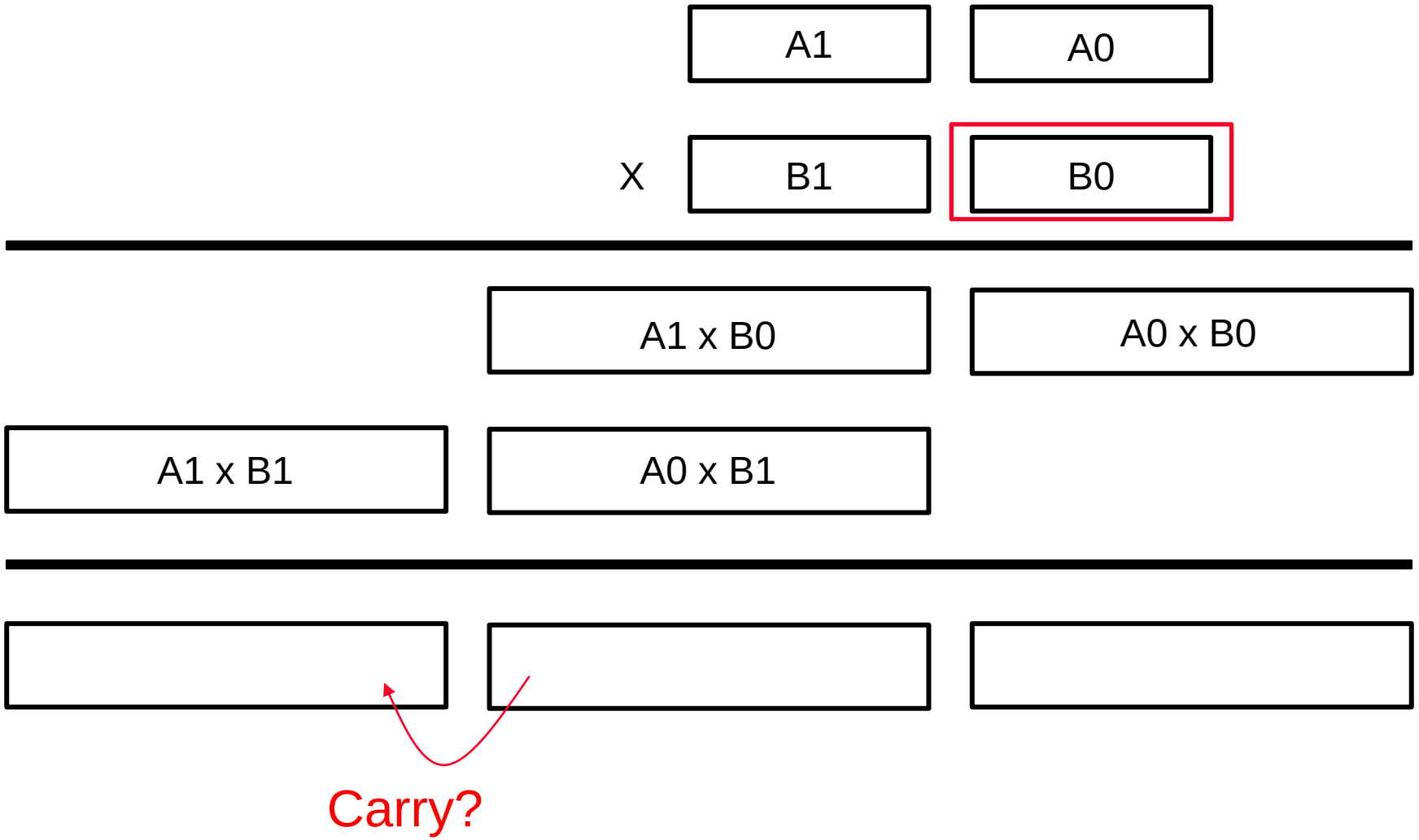
  $g = g_0.2^0 + g_1.2^{32} + g_2.2^{64} + g_3.2^{96} + g_4.2^{128} + g_5.2^{160}$
- **Multiplying f and g term by term**

Partial Accumulators Method (Cont.)

$2^{192} \equiv 2^{64} + 1 \pmod{P192}$

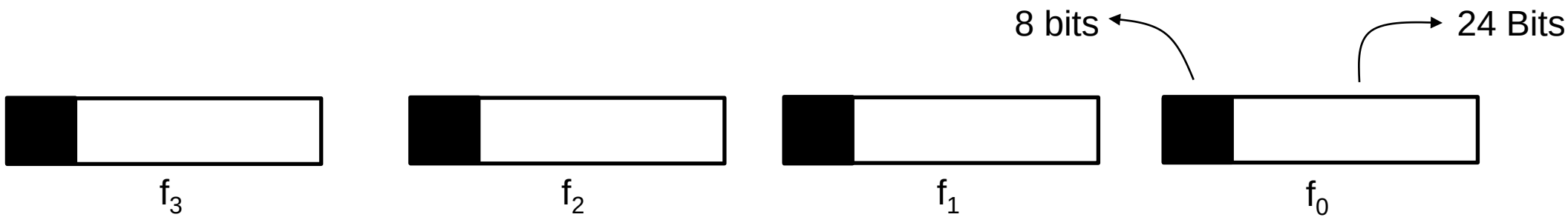


Multi-Precision Multiplication

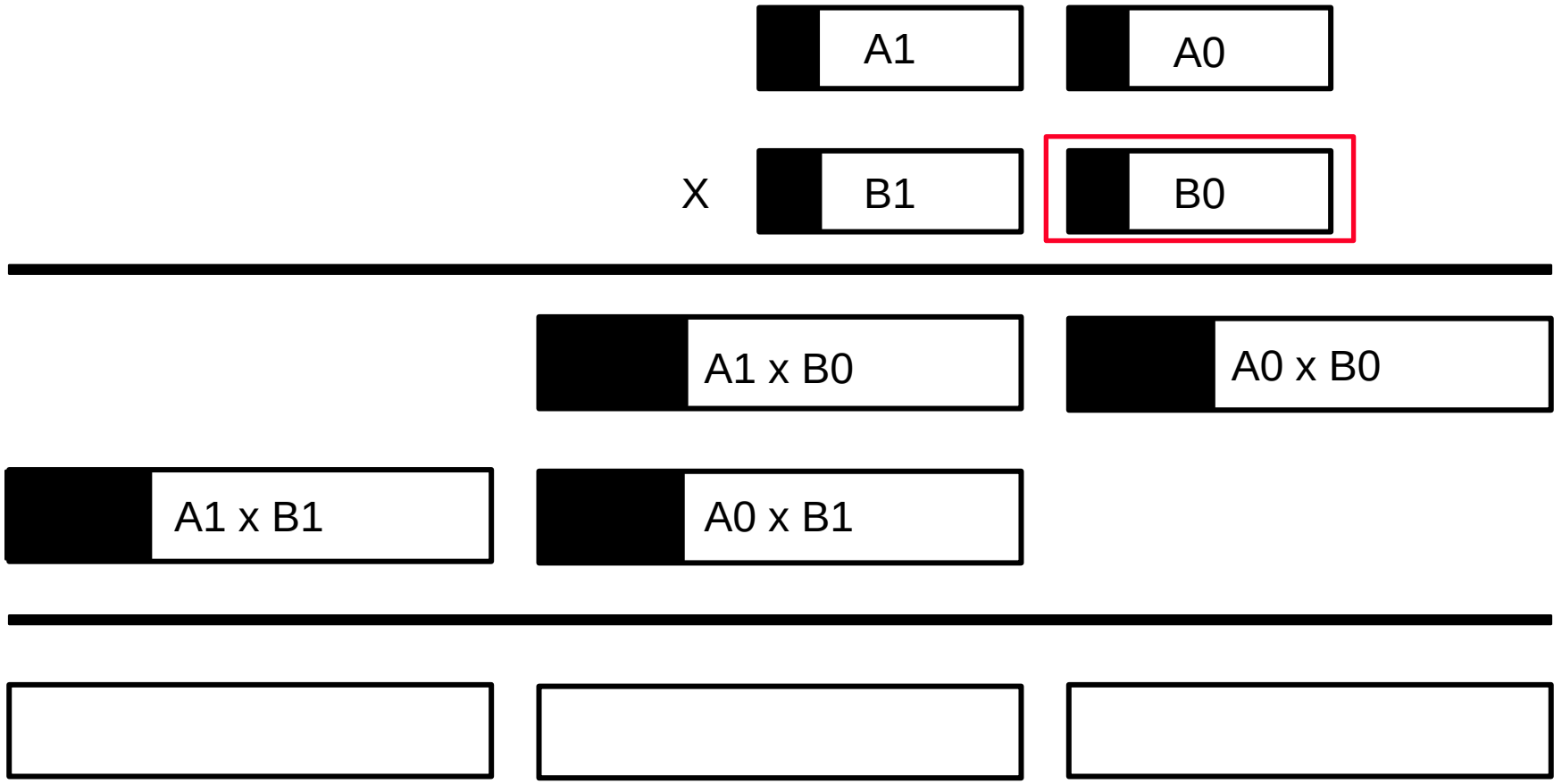


Redundant representation

- **Number is represented such that there is enough space for carry bits**
- $f = f_0.2^0 + f_1.2^{24} + f_2.2^{48} + f_3.2^{72} + f_4.2^{96} + f_5.2^{120} + f_6.2^{144} + f_7.2^{168}$



Multi-Precision Multiplication (Cont')

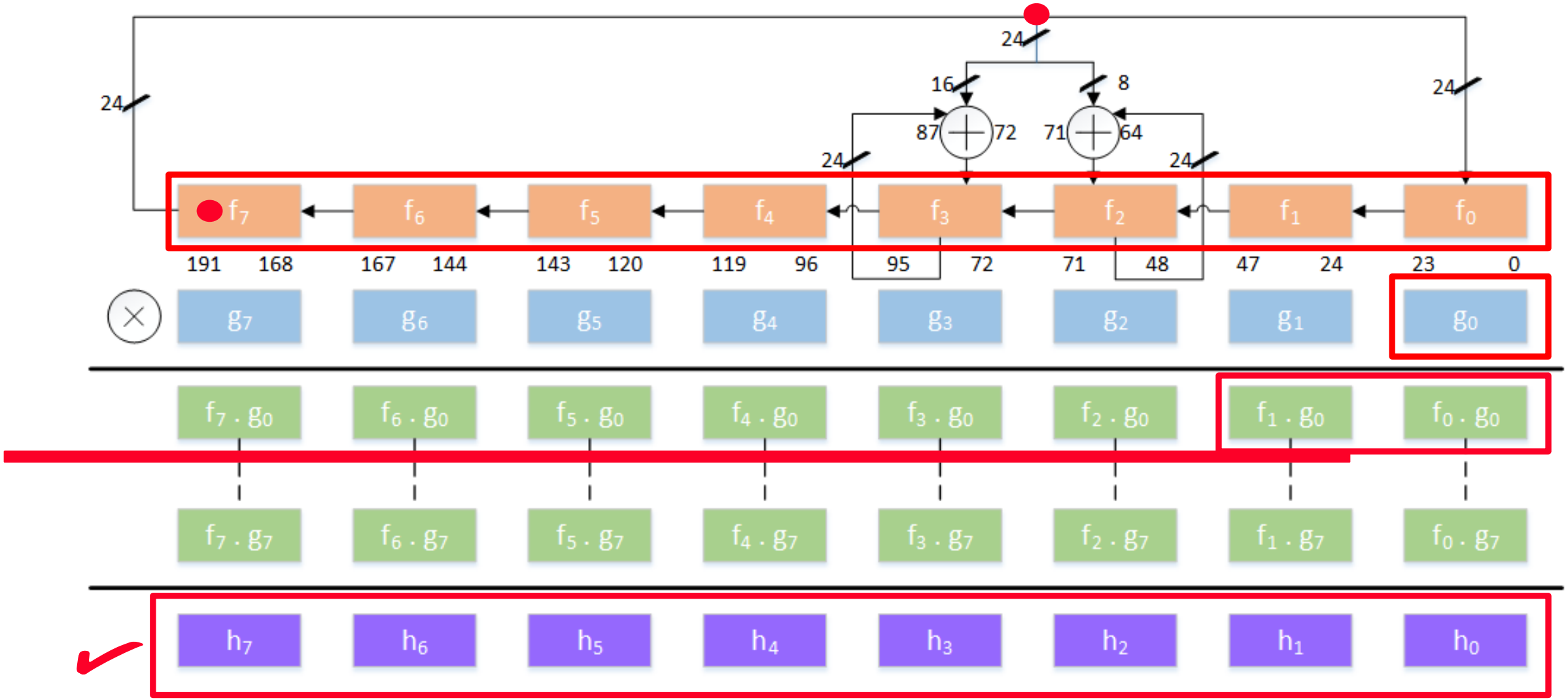


No carry

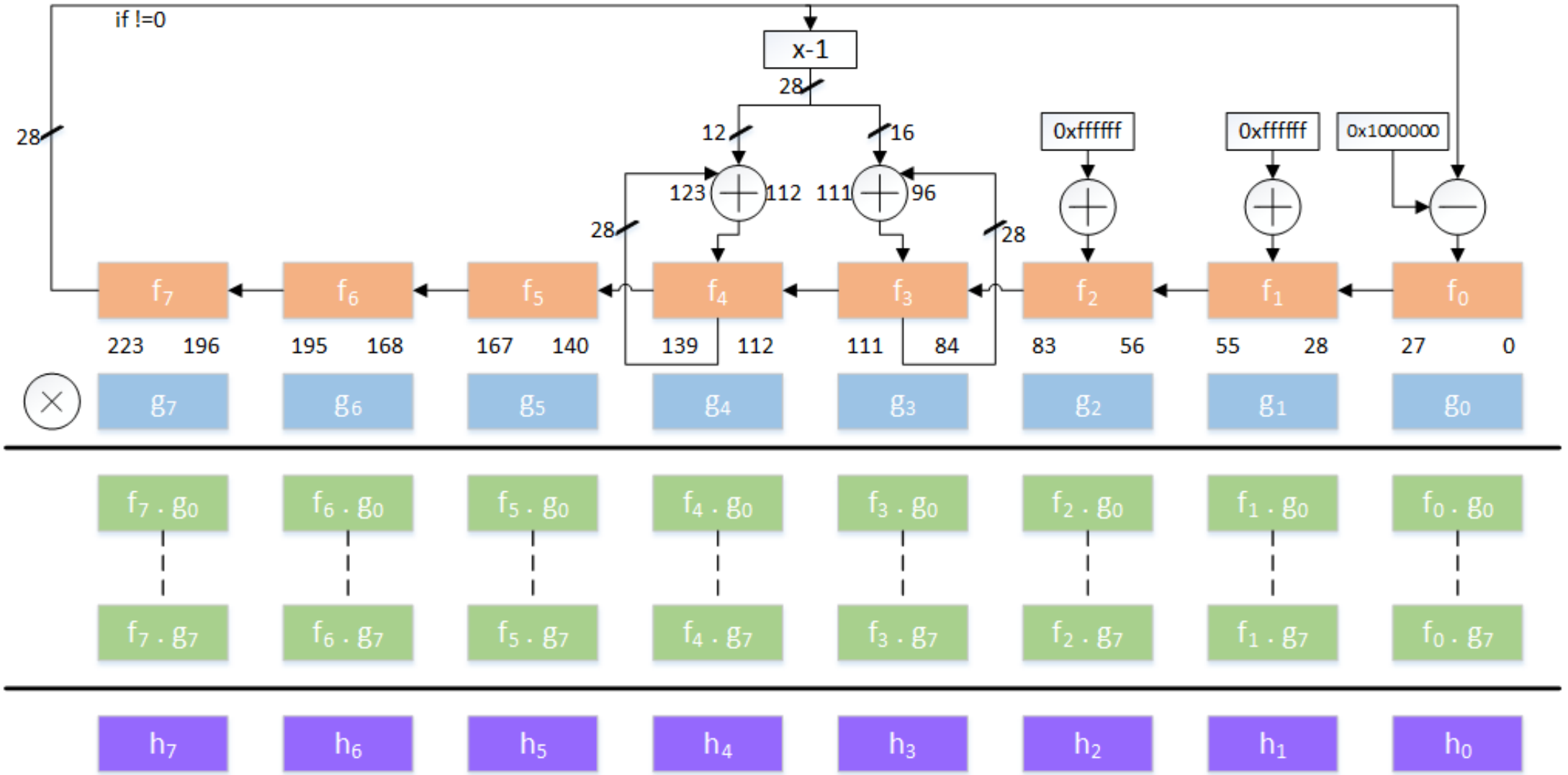
Multiplicand Reduction Method

- **Instead of reducing the partial products, reduce the multiplicand at each step**
- **f and g represented in redundant form**

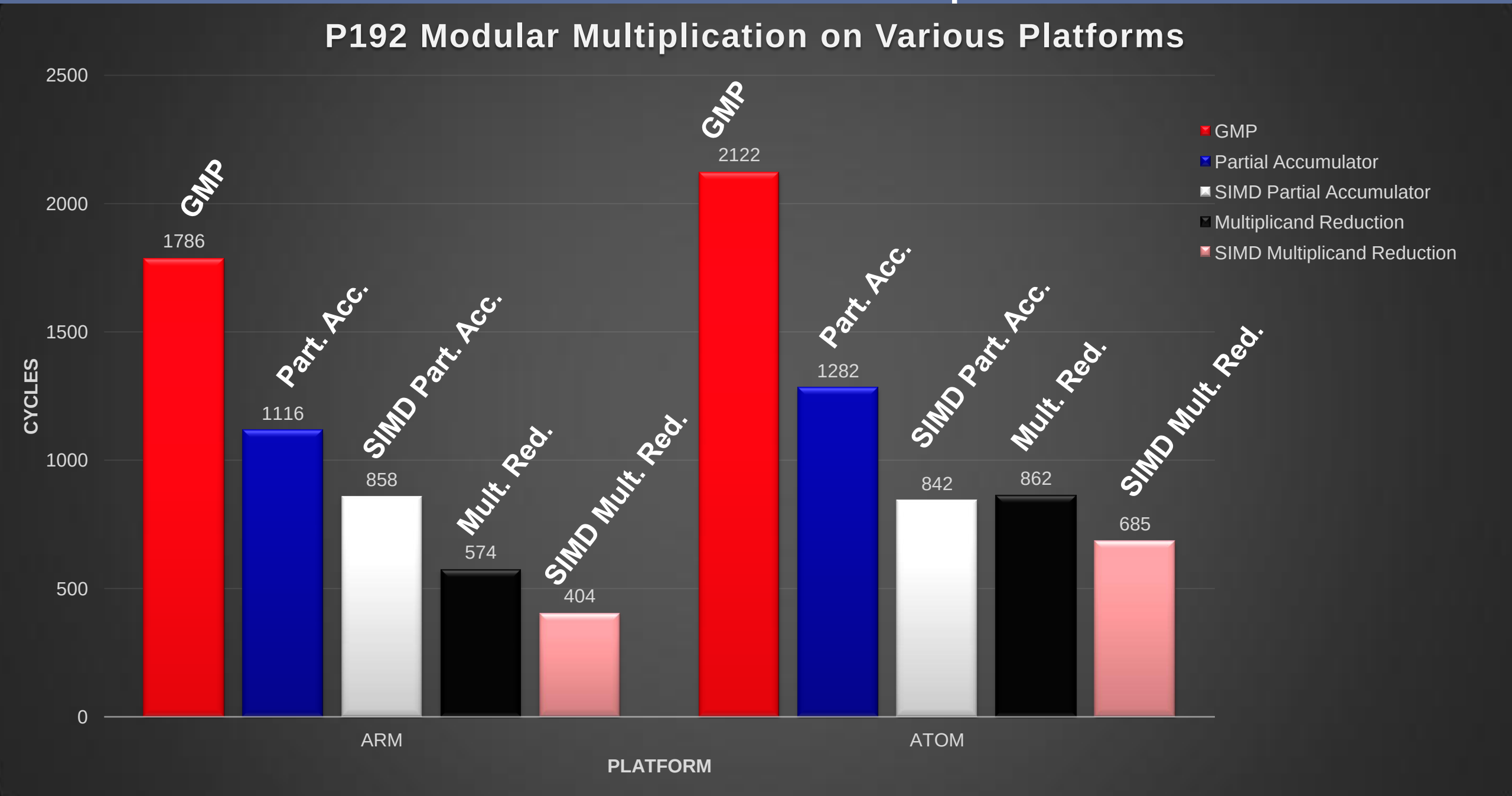
P192 = $2^{192} - 2^{64} - 1$



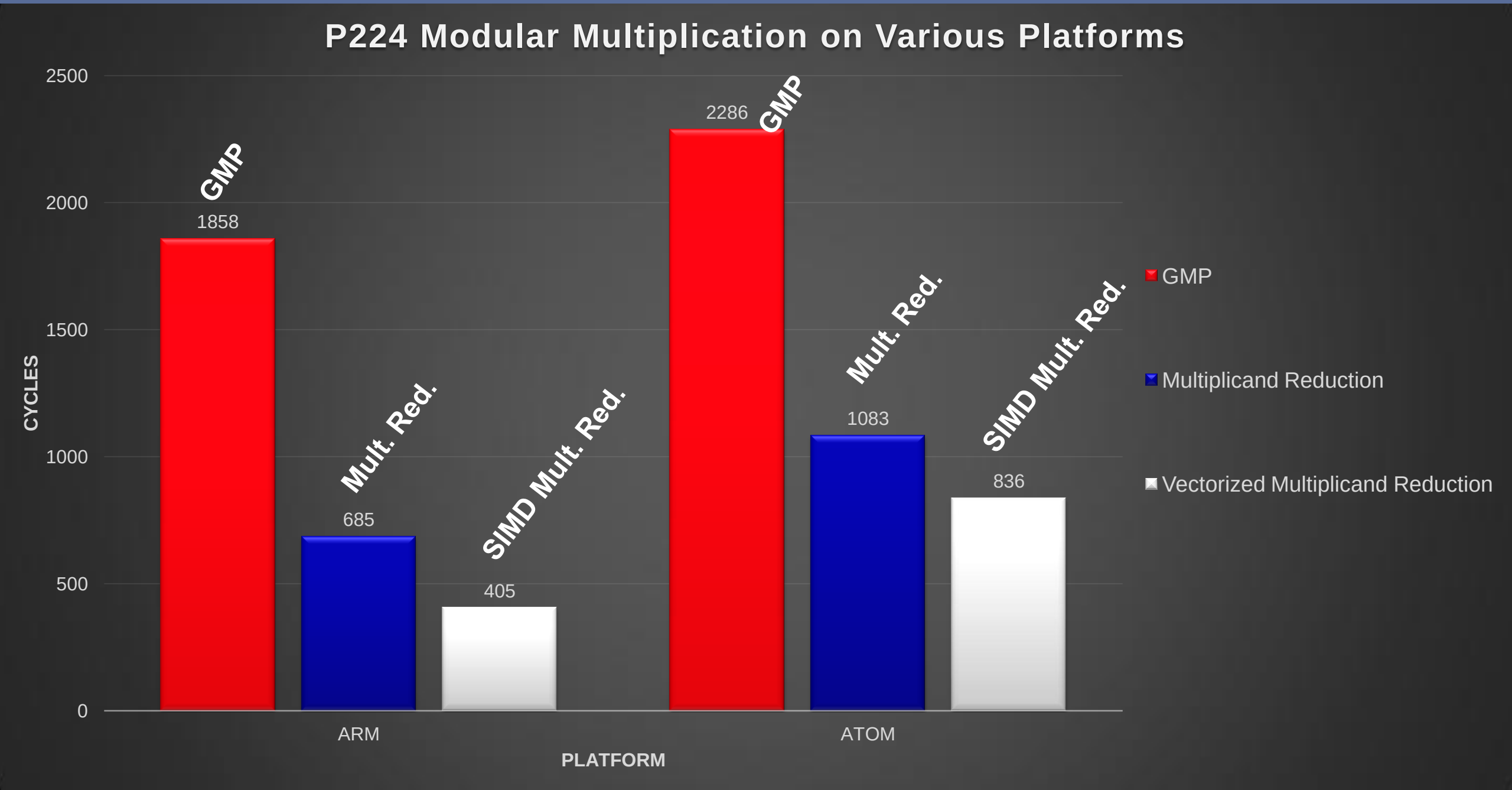
$P224 = 2^{224} - 2^{96} + 1$



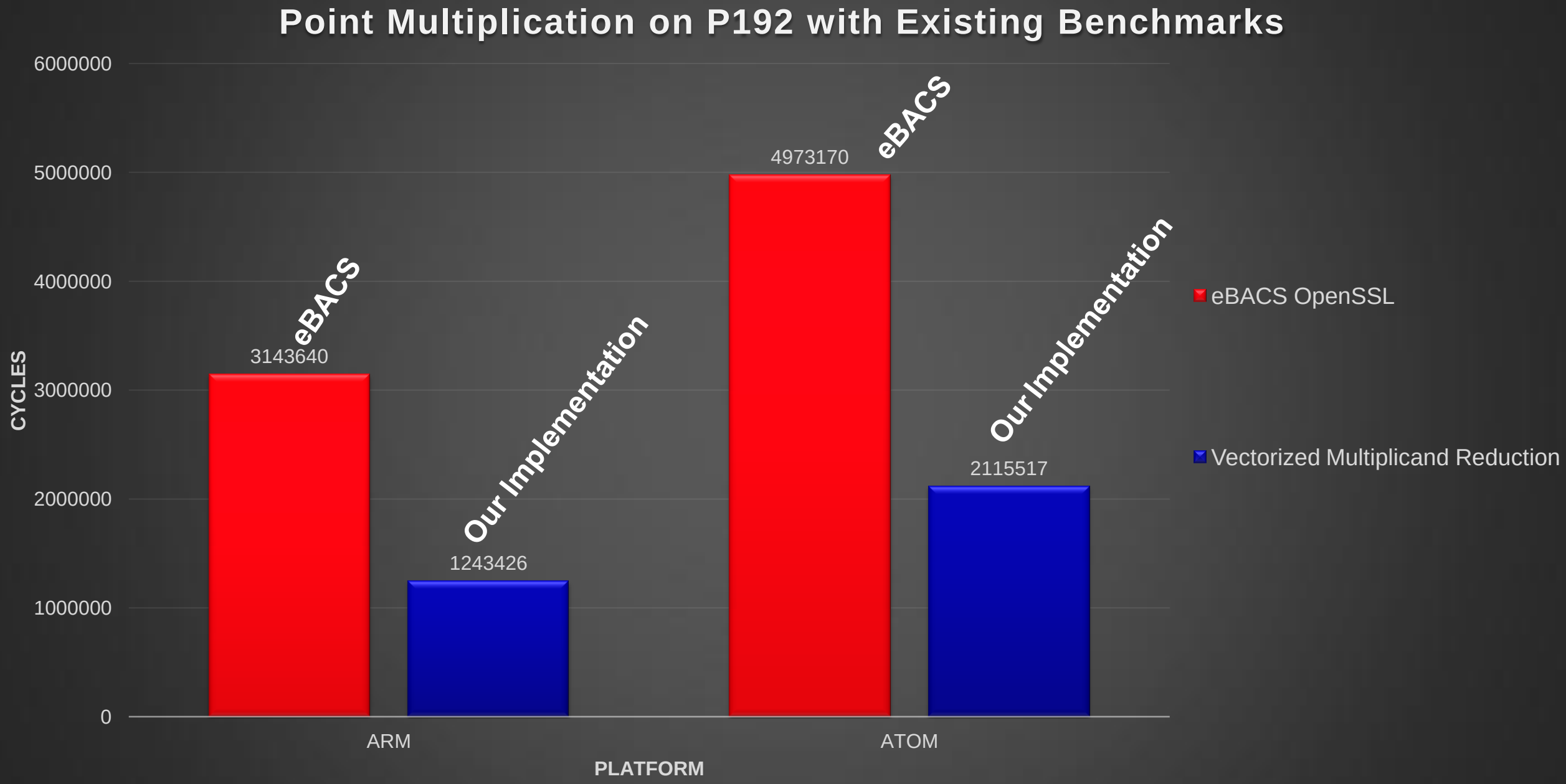
Comparison of Results



Comparison of Results (Cont')



Comparison of ECC Point Multiplication Cycles



Conclusion

- **Both the approaches are significantly faster than GMP on both the platforms**
- **The vectorized versions run faster than non vectorized versions**
- **ECC point multiplication for P192 is more than two times faster than the existing benchmarks from eBACS**
- **Future Work:**
  - **More optimizations can be done by vectorizing final accumulation**

QUESTIONS??