저전력, 저면적, 고성능 메르센 소수 모듈러 연산 하드웨어 구현

A Low-Power, Area-Efficient, and High-Performance Hardware Implementation of Mersenne Prime Modular Arithmetic

2025 전자공학과 산학박람회

Researcher 조성원, 이재문, 조민준 **Professor** 이종민

ABSTRACT

인공지능 및 사물인터넷의 발전으로 데이터 처리 속도와 보안의 중요성이 증가함에 따라 공개키 암호 알고리즘에서의 모듈러 연산 효율이 중요한 과제로 부상하고 있다.

본 연구는 2^n-1로 표현되는 메르센 소수의 수학적 특성에 착안하여 복잡한 나눗셈을 단순 덧셈과 시프트(Shift) 연산으로 대체하여 연산 과정을 획기적으로 경량화하였다.

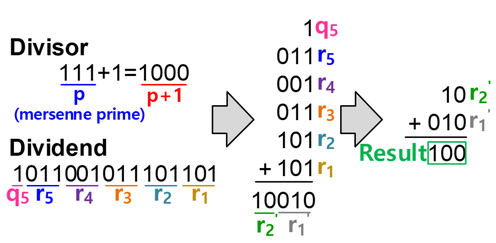
또한, Pipeline 구조, 병렬 연산, 트리 구조 비교 알고리즘 등 최적화를 통해 연산 지연을 감소시키고 전력 및 면적 소모량을 개선하여, 저전력, 저면적, 고성능 메르센 모듈러 연산을 180nm CMOS 공정에서 하드웨어로 구현하였다.

OBJECTIVES

본 연구는 모듈러 연산 과정에서 발생하는 비교 연산과 뺄셈 연산을 개선하기 위해 2^n-1로 표현되는 메르센 소수의 특성을 활용하여 기본적인 % 연산자 기반 모듈러 연산기 대비 저전력, 저면적, 고성능 메르센 소수 모듈러 연산기의 하드웨어 구현을 목표로 한다.

METHODOLOGY

A) 메르센 소수 기반 모듈러 연산 알고리즘



$$
\begin{aligned}
n &= pq + r \\
&= (p+1)q_1 + r_1 \\
&= (p+1)((p+1)q_2 + r_2) + r_1 \\
&= \cdots \\
&= (p+1)^k q_k + (p+1)^{k-1} r_k + \cdots + r_1 \\
&= p \cdot f(p) + q_k + \sum_{i=1}^{k} r_i
\end{aligned}
$$

$(n \in \mathbb{Z}, p: Mersenne\ prime\ number, q: Quotient, r = Remainder)$

핵심 원리

1. 피제수(Dividend)를 메르센 소수(Divisor)의 비트 길이(p)로 나누어 상위 비트(Upper bits)와 하위 비트(Lower bits)로 분할.
2. Upper bits + Lower bits 덧셈 연산 수행.
3. 덧셈 결과가 여전히 메르센 소수보다 클 경우, 결과값에 대해 1~2 과정을 결과값이 메르센 소수보다 작아질 때까지 반복.

B) OPTIMIZATION

a. 병목 구간 최적화를 위한 병렬 구조 도입

Algorithm 1 Parallel Structure

```
for i = 0 to m − 1 do
    lt_block[i] ← (next_accum[i ∗ n+ : n] ≤ bit_mask[i ∗ n+ : n])
end for
```

입력 데이터의 길이가 증가하면 단일 비교기를 사용한 연산은 시스템의 Critical Path가 되어 성능을 저하시킨다. 이러한 병목현상을 해결하기 위해, 긴 데이터 벡터를 여러 개의 n-bit 블록으로 분할하고, 다수의 비교기를 이용해 각 블록을 동시에 병렬적으로 비교하는 구조를 설계했다. 이를 통해 비교 연산의 지연 시간을 크게 단축시켜 전체 처리 속도를 향상시킨다.

b. 트리 구조 비교 알고리즘

Algorithm 2 Tree-Based Completion Detection

```
// Check if modular reduction is complete
for i = 0 to 15 do
    lt_level1[i] ← lt_block[2i] & lt_block[2i + 1]
end for
for i = 0 to 7 do
    lt_level2[i] ← lt_level1[2i] & lt_level1[2i + 1]
end for
for i = 0 to 3 do
    lt_level3[i] ← lt_level2[2i] & lt_level2[2i + 1]
end for
for i = 0 to 1 do
    lt_level4[i] ← lt_level3[2i] & lt_level3[2i + 1]
end for
reduction_complete ← lt_level4[0] & lt_level4[1]
```

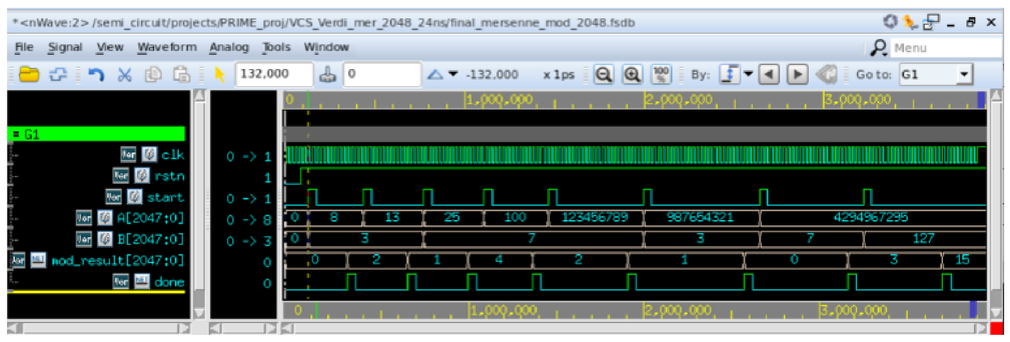
앞서 도입한 병렬 비교 구조는 다수의 비교 결과 플래그를 생성한다. 이 플래그들을 하나의 최종 완료 신호로 통합하는 과정에서, 이를 순차적으로 처리한다면 m개의 블록 수에 비례하는 O(m)의 시간 복잡도를 갖게 된다.

이 문제를 해결하기 위해 2-input AND 게이트를 이진 트리 형태로 배치하여 모든 플래그를 $O(\log_2 m)$의 시간 복잡도로 빠르게 통합하는 방식을 적용한다. 이러한 트리 구조는 완료 신호 생성까지의 지연을 최소화하여 FSM이 다음 상태로 신속하게 전이하도록 함으로써, 연산 사이클을 단축시킨다.

RESULT

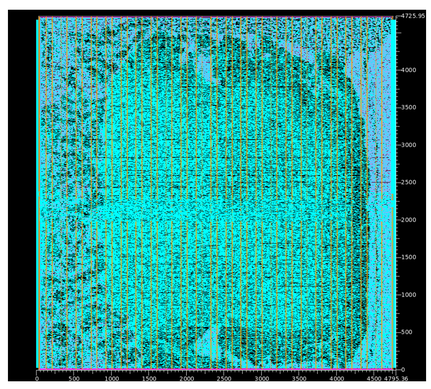
A) 구현 환경 및 Post Simulation 결과

- 공정 : 180nm TSMC CMOS 공정
- 설계 Tool : Synopsys EDA Tools (VCS/Verdi, Design Compiler, PrimeTime, Formality, IC Compiler)
- 구현 내용 : 최대 2048-bit 입력 데이터 길이까지 지원하는 메르센 소수 모듈러 연산기 구현



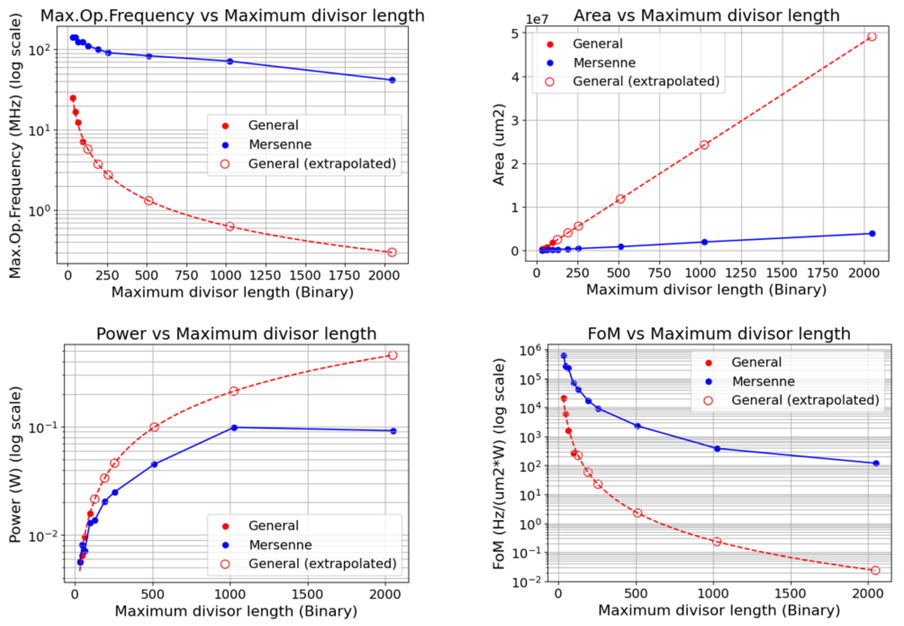
최종적으로 P&R까지 진행 후 Post Simulation 결과, 정상 동작을 확인하였다.

B) Layout 결과



IC Compiler를 이용하여 2048-bit 메르센 소수 기반 모듈러 연산기의 하드웨어 Implementation을 진행한 결과이다. (Area = $3.83mm^2$)

C) 일반 모듈러 연산기와의 성능 비교



성능, 소비 면적, 소비 전력을 종합적으로 평가할 수 있는 지표 (Figure of Merit)

$$FoM = \frac{Max.Op.Frequency}{Area \times Power}$$

32 – 2048비트 입력 데이터 길이에 대한 최대 동작 주파수, 소비 전력, 사용 면적 측정 결과, 일반 모듈러 연산기 대비, 약 5.7-138.9배의 성능 향상, 5.2-14.5배의 면적 효율 향상, 0.8-4.9배의 전력 효율 향상을 달성하였다.

CONCLUSIONS

본 연구에서는 메르센 소수의 수학적 특성을 활용하여 메르센 소수 전용 모듈러 연산기를 구현하였다. 메르센 소수 기반 모듈러 연산을 위해 반복적으로 수행되는 비교기를 병렬 및 트리 구조를 활용하여 최적화하였다.

180nm CMOS 공정으로 구현한 메르센 소수 기반 모듈러 연산기는 일반 모듈러 연산기 대비 약 5.7-138.9배의 성능 향상, 5.2-14.5배의 면적 효율 향상, 0.8-4.9배의 전력 효율 향상을 달성하였다. 또한, 성능, 소비 면적, 소비 전력을 통합적으로 고려하여 평가한 결과, 30.3-5000배 높은 FoM을 달성하였다.

이러한 결과는 제안하는 메르센 소수 기반 모듈러 연산기가 고속 암호 연산을 위한 효율적인 구조임을 입증한다.

아주대학교 | 전자공학과 KIAT 반도체특성화대학사업단 GUIP 경기도 대학혁신플랫폼