정보보호학술대회 영남지부

WebAssembly로 구현한 CHAM

한성대학교
안규황, **권혁동**, 김현준, 서화정

HSU HANSUNG UNIVERSITY

https://github.com/kyu-h/WebAssembly_CHAM

목 차

1. Assembly 소개

2. Web Assembly 소개

3. 성능 평가 (Web Assembly CHAM vs Javascript CHAM)

4. 결론

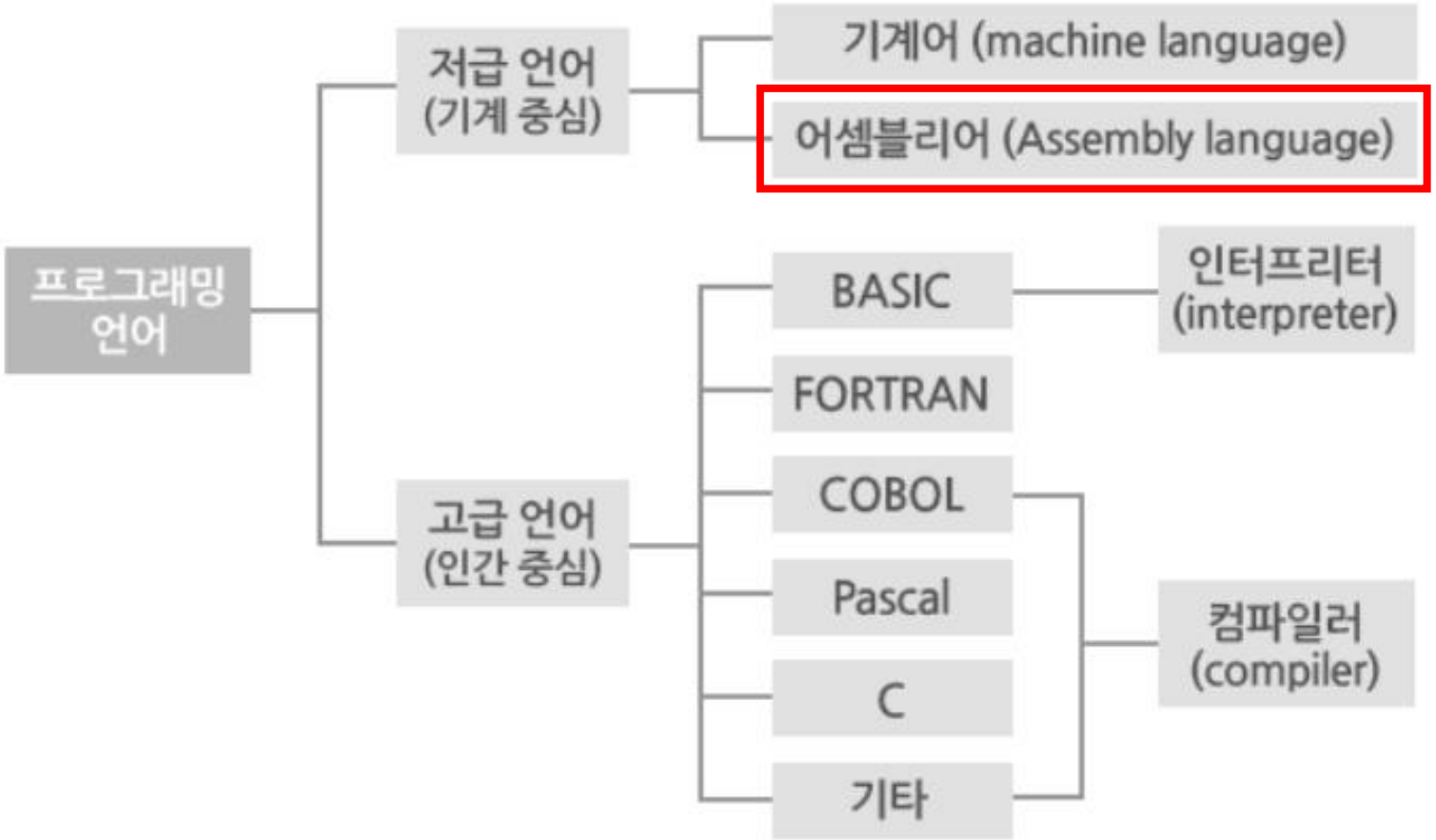
HSU
HANSUNG
UNIVERSITY

Assembly

HANSUNG UNIVERSITY

프로그래밍 언어

프로그래밍 언어는 크게 기계 중심 혹은 인간 중심으로 나누어짐



HSU HANSUNG UNIVERSITY

어셈블리어란?

- 0과 1의 이진수로 프로그래밍을 하는 기계어는 컴퓨터가 바로 읽을 수 있다는 점 빼면 사용하기 아주 불편
- 이를 기계어보단 느리지만 인간이 그나마 이해할 수 있는 언어로 표현한 것을 어셈블리라 칭함
- 하드웨어와 스프트웨어의 가장 밑바닥에 있는 저급 언어

어셈블리어 특징

- 명령 실행 속도가 아주 빠름
- 매우 세밀한 프로그래밍 스킬이 필요
- 어셈블리어는 하드웨어 특성을 탄다
  → 하드웨어 종류에 따라 사용하는 어셈블리어도 달라짐

HSU HANSUNG UNIVERSITY

어셈블리어 장·단점

장점

- 프로그램의 실행 속도가 아주 빠름
- 프로그램의 크기가 매우 작음
- 어떤 기계에서도 사용 가능

단점

- 배우기 어려움
- 큰 프로그램을 구성하기 어려움
- 하드웨어별로 사용하는 어셈블리가 다름
  → 새로 배워야 함

HSU HANSUNG UNIVERSITY

Web Assembly

HANSUNG UNIVERSITY

웹어셈블리(WebAssembly, WASM)란?

- 구글, 마이크로소프트, 애플, 모질라가 소속된 웹 어셈블리 커뮤니티 그룹이 웹 성능을 향상하기 위해 2015년부터 개발한 웹브라우저 표준 포맷

- 웹 브라우저를 돌릴 수 있는 새로운 형식의 코드로 2017년 3월부터 모든 웹 브라우저에 도입

- 기존에는 javascript를 이용하여 웹을 구성했다면,
  웹어셈블리는 C/C++/Rust 등을 이용해 웹 페이지 구현 가능

브라우저별 웹 어셈블리 제공 최소 버전

Chrome	57
Edge	16
Firefox	52
Explorer	Not Support
Safari	11

HSU HANSUNG UNIVERSITY

웹 어셈블리 vs 자바스크립트

- 웹 어셈블리가 나오기전까지 웹 페이지를 표현하는데 사용할 수 있는 언어는 자바스크립트 밖에 존재하지 않았음

- 개발 언어에는 Low-Level 언어와 High-Level 언어가 존재함

  → Low-Level 언어는 컴퓨터가 그나마 이해하기 쉬운 언어 ex) C 언어
  High-Level 언어는 인간이 그나마 이해하기 쉬운 언어 ex) 자바스크립트

- 따라서, High-Level 언어일 수록 컴퓨터가 이해하기 힘들기 때문에 컴파일 하는데 오랜 시간이 걸림

HSU
HANSUNG
UNIVERSITY

웹 어셈블리 vs 자바스크립트

- 웹 어셈블리는 C/C++로 구현한 것을 선형메모리 상에서 '.wasm' 이라는 바이너리 파일로 컴파일한 후 웹 페이지 내 'glue' 라는 JS 영역에 이식하여 웹 페이지를 구성

- 웹 어셈블리가 개발됨에 따라 기존에 자바스크립트로 구현한 것 보다 상당한 속도로 웹 페이지를 구현할 수 있게 됨

- 웹 어셈블리를 이용하여 암호를 구현하여 이를 검증하기 하고자 함

HSU HANSUNG UNIVERSITY

웹 어셈블리로 왜 암호 구현을?

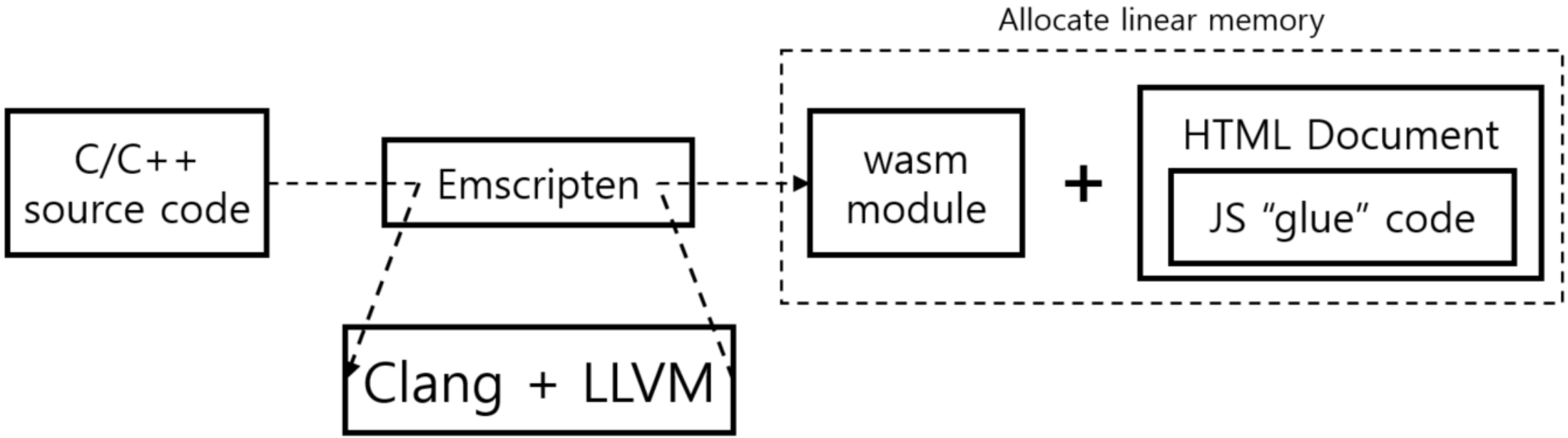
- 병렬화 프로그래밍을 할 수 있는 플랫폼인 OpenMP의 경우,
  → 암호 쪽에서 활용되기 위해서 만들어진 플랫폼이 아닌 데이터 시각화, 영상처리 쪽에서 좀더 빠르게 사용하기 위해 만들어진 플랫폼임

  - 하지만 현재는 암호 구현 쪽에서 많이 활용되고 있음

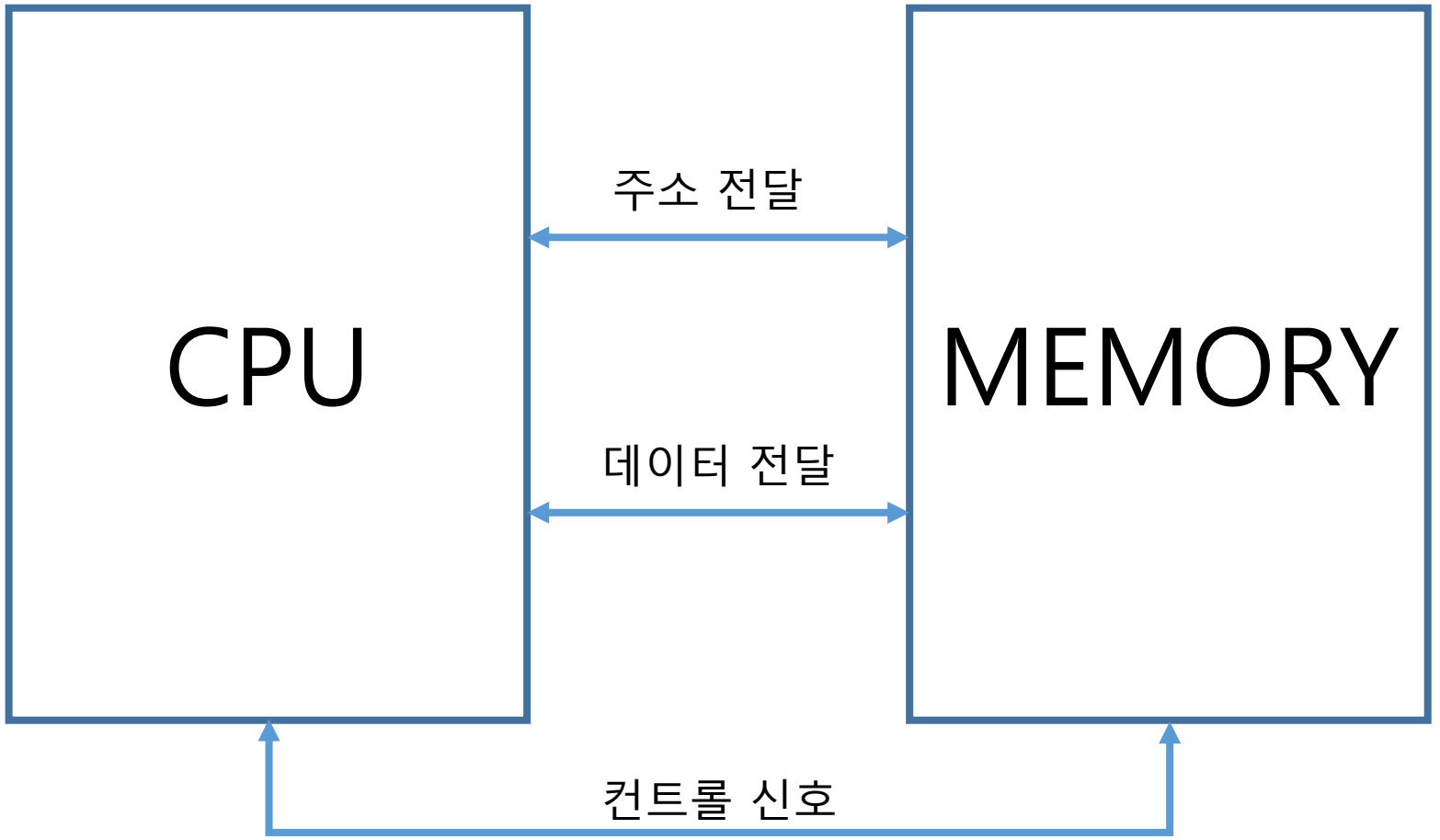
- 웹 어셈블리 역시 암호를 위해 개발 된 플랫폼이 아님
  → 그러나 위의 예와 같이 암호 쪽에서 충분히 활용될 수 있는 여지가 보임, 구현 속도의 향상으로 이득을 취할 수 있을 것 같다는 예상

HSU HANSUNG UNIVERSITY

웹 어셈블리 구조

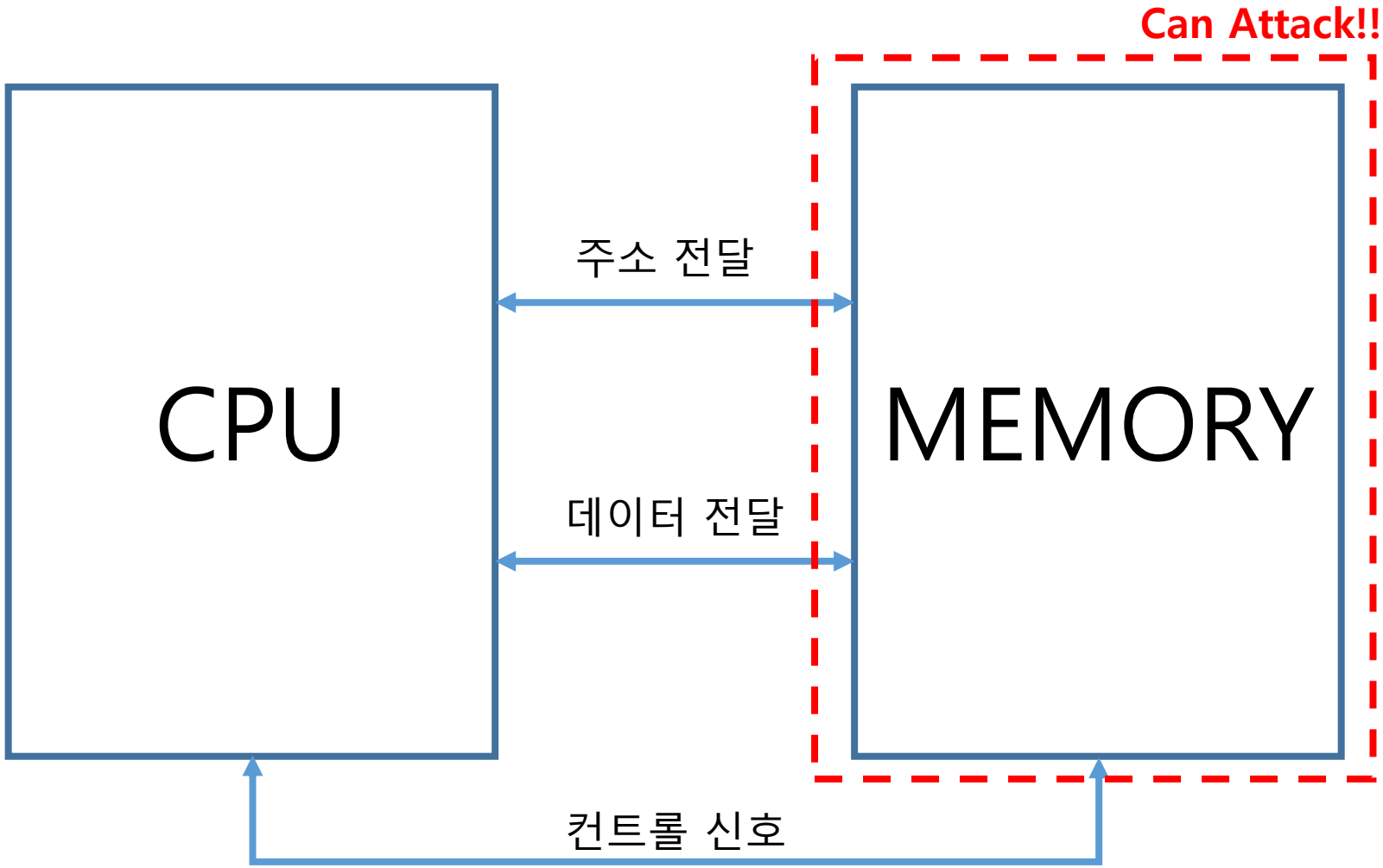

Allocate linear memory
C/C++
source code
Emscripten
Clang + LLVM
wasm
module
+
HTML Document
JS "glue" code

기존 메모리 구조



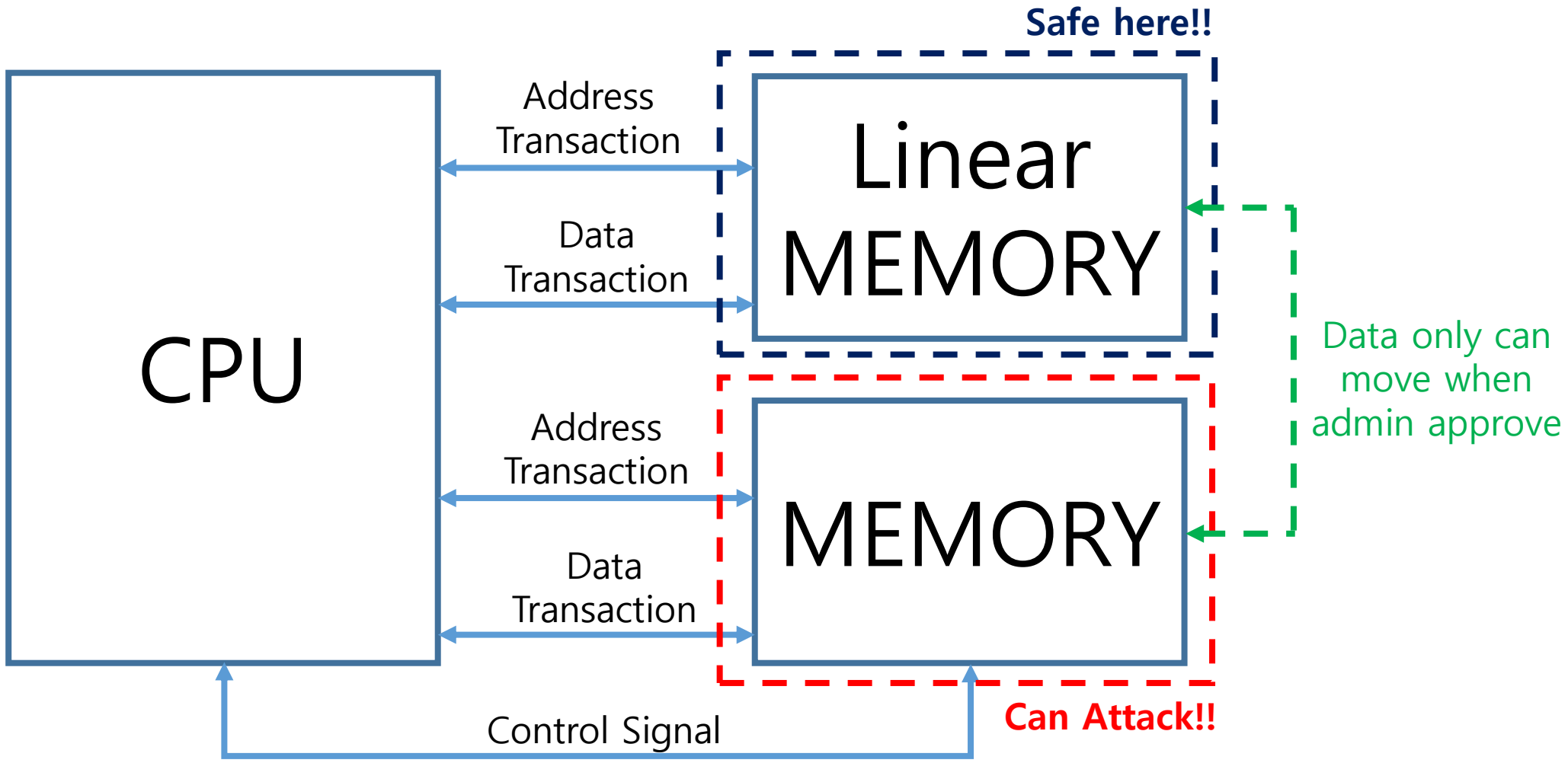
HSU
HANSUNG
UNIVERSITY

기존 메모리 접근 구조



HSU HANSUNG UNIVERSITY

웹어셈블리 메모리 접근 구조



HSU HANSUNG UNIVERSITY

성능 비교

HANSUNG UNIVERSITY

웹 어셈블리 & 자바스크립트를 이용한 AES 구현

구현 환경

OS	**MacOS 10.12.6**
Processor	Intel Core i5 2.7 GHz
IDE	Brackets 1.10
Web Browser	Chrome 69.0.3497.100

HSU HANSUNG UNIVERSITY

https://github.com/kyu-h/WebAssembly_CHAM

웹 어셈블리 & 자바스크립트를 이용한 CHAM 구현

최대 약 6.15배 성능 향상

Language	Time(s)	Cpb
1회 암호화 하는데 수행된 퍼포먼스		
Web Assembly	0.0013	62,678
Javascript	0.008	385,714
100회 암호화 하는데 수행된 퍼포먼스		
Web Assembly	0.0083	400,178
Javascript	0.051	2,458,928
10,000회 암호화 하는데 수행된 퍼포먼스		
Web Assembly	0.0873	4,209,107
Javascript	0.259	12,487,500

HSU HANSUNG UNIVERSITY

웹 어셈블리 & 자바스크립트를 이용한 AES 구현

최대 약 3.5배 성능 향상
그러나 부여 된 선형 메모리(Linear Memory) 영역을 초과할 경우 오버헤드 발생

Language	Time(s)	Cpb
1회 암호화 하는데 수행된 퍼포먼스		
Web Assembly	0.0031	116,250
Javascript	0.011	412,500
100회 암호화 하는데 수행된 퍼포먼스		
Web Assembly	0.0218	817,500
Javascript	0.068	2,550,000
10,000회 암호화 하는데 수행된 퍼포먼스		
Web Assembly	2.0848	78,180,000
Javascript	0.619	23,212,500

HSU HANSUNG UNIVERSITY

결 론

HSU HANSUNG UNIVERSITY

결 론

- 웹 어셈블리로 구현 한 CHAM이 자바스크립트로 구현한 CHAM보다 최대 약 6.15배 빠름
- 웹 어셈블리로 구현할 경우 보안성도 강화 됨
- 단, 웹 어셈블리를 구현하기 위해 할당 된 선형 메모리 범주를 초과해서는 안됨 → 오버헤드 발생 → 선형 메모리 범주를 늘려주면 해결 가능

HSU HANSUNG UNIVERSITY

https://github.com/kyu-h/WebAssembly_CHAM

Thank You!

HANSUNG UNIVERSITY

https://github.com/kyu-h/WebAssembly_CHAM