Privacy-Preserving Computation

Always encrypted processing

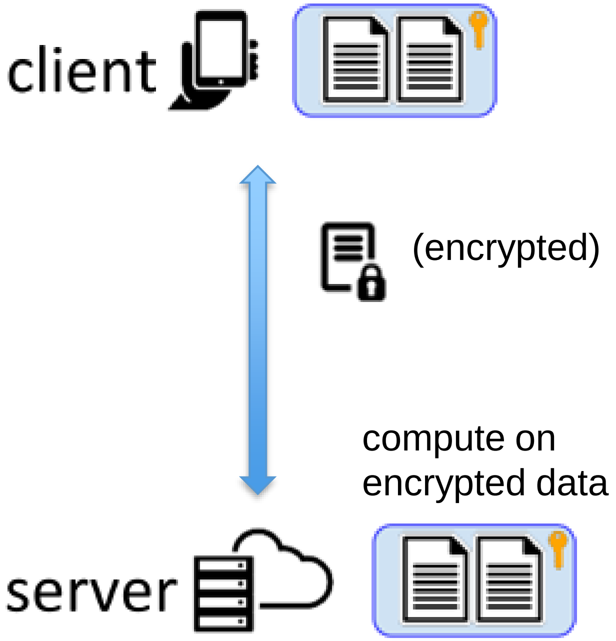
- Communicate, store, and compute with **encrypted** data

Schemes for privacy-preserving Computation

- Homomorphic Encryption
- Secret Sharing
- Garbled Circuits

Garbled Circuits Support Arbitrary Computation

- Arithmetic and Boolean logic
- Conditionals (e.g., ReLU in deep learning)



Garbled Circuits (GCs)

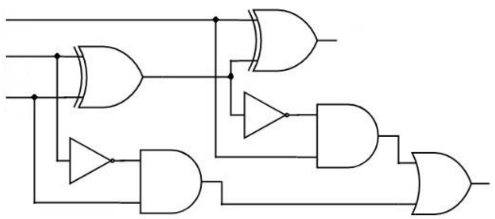
Garbling Phase

```
f (a, b):
    return (a>b)
```

(0, 1)

Secret Labels

$(Xi, i \in 0,1)$



🔒 garbled circuit (Secret Gates)

encrypted input to encrypted gate

Generator
Garbler
(Alice)

garbled circuit *f*

garbled input 🔒 *X*

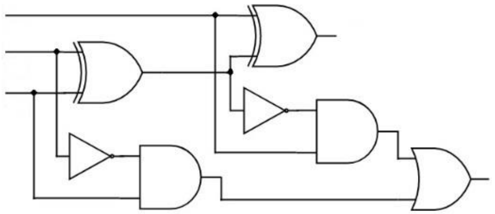
🔒 *f* (X, Y)

f (x, y)

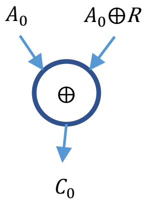
Evaluator
Eval
(Bob)

Evaluating Phase

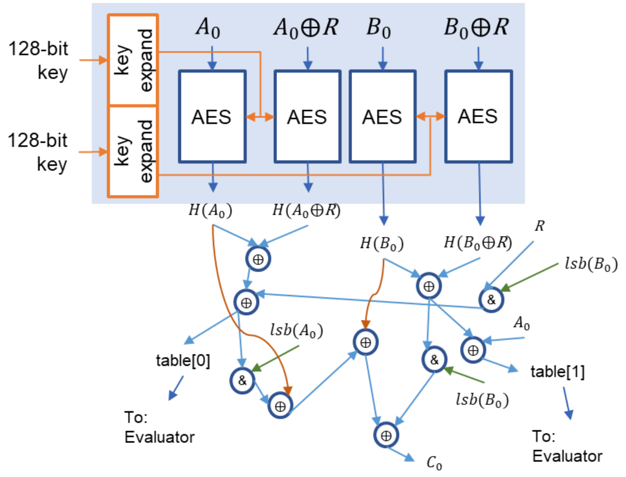
garbled label *X*
garbled label *Y*



🔒 *f* (X, Y)



XOR in GCs

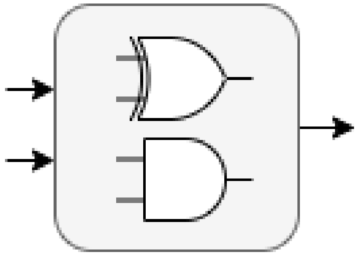


AND in GCs

Applications Driving Architectures

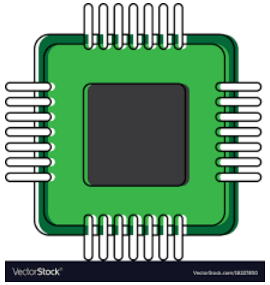
HAAC: A Garbled Circuits Half-Gate Accelerator

Custom Logic



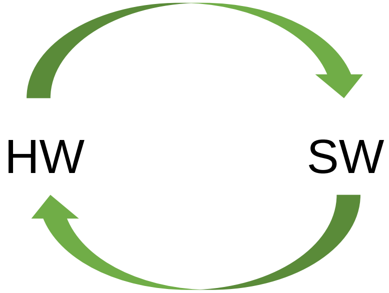
Speeds up GCs gate computations by 153.8 ×

Architecture



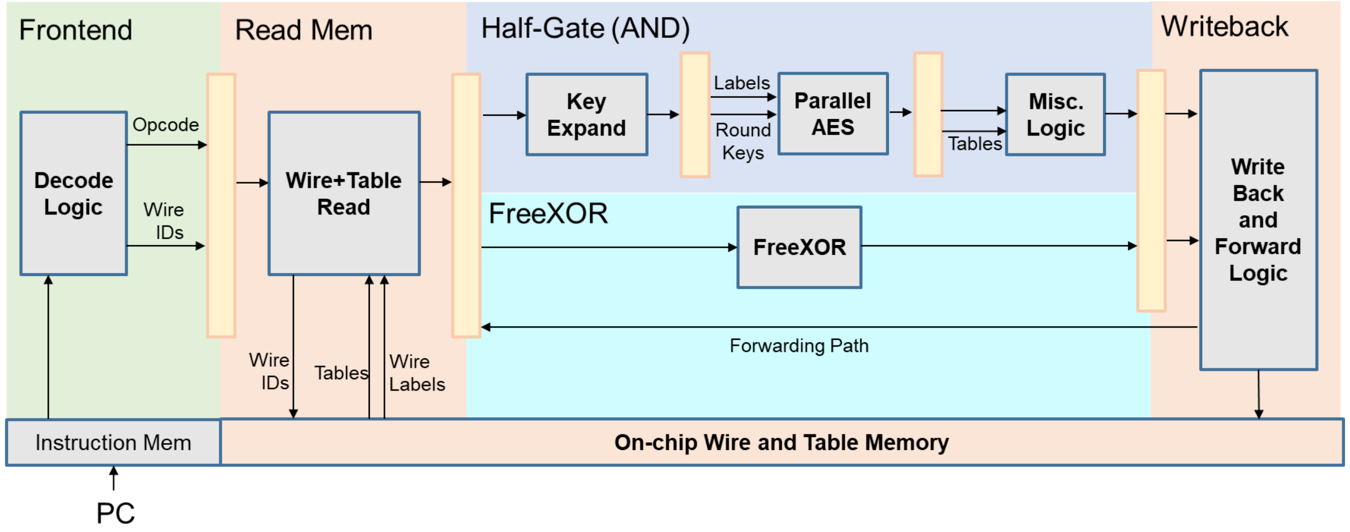
Parallel gate processing, provides additional 13.7 × speedup

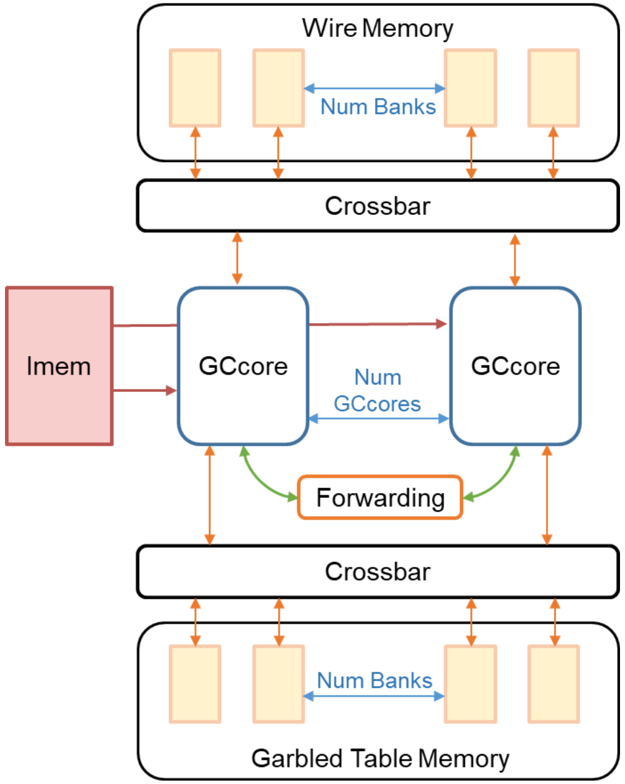
Compiler



Automatic programming, performance optimizations, eliminating data dependence and bank conflicts

Hardware Architecture



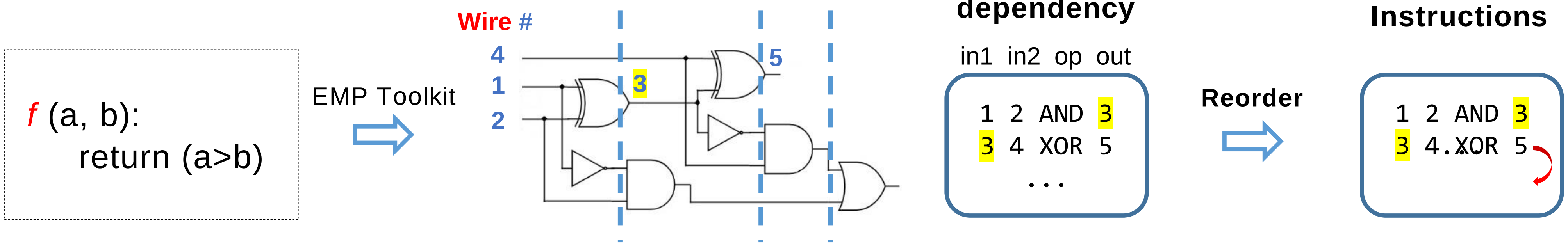


- Pipeline: 18 stages for Garbler Half-Gate
- A small Forwarding logic enables fast data reuse

- Multi-core performs instruction level parallelism
- 1 MB on-chip memory, multi-bank improves memory access

Compiler – Reorder & Rewire

- Resolve Gates Dependencies

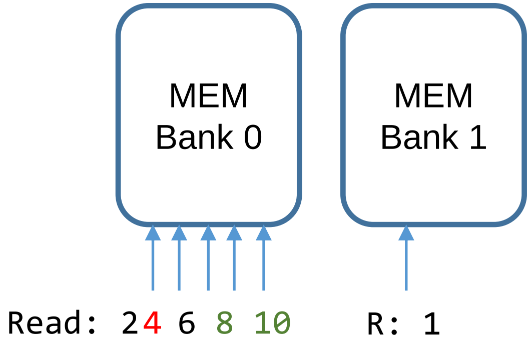


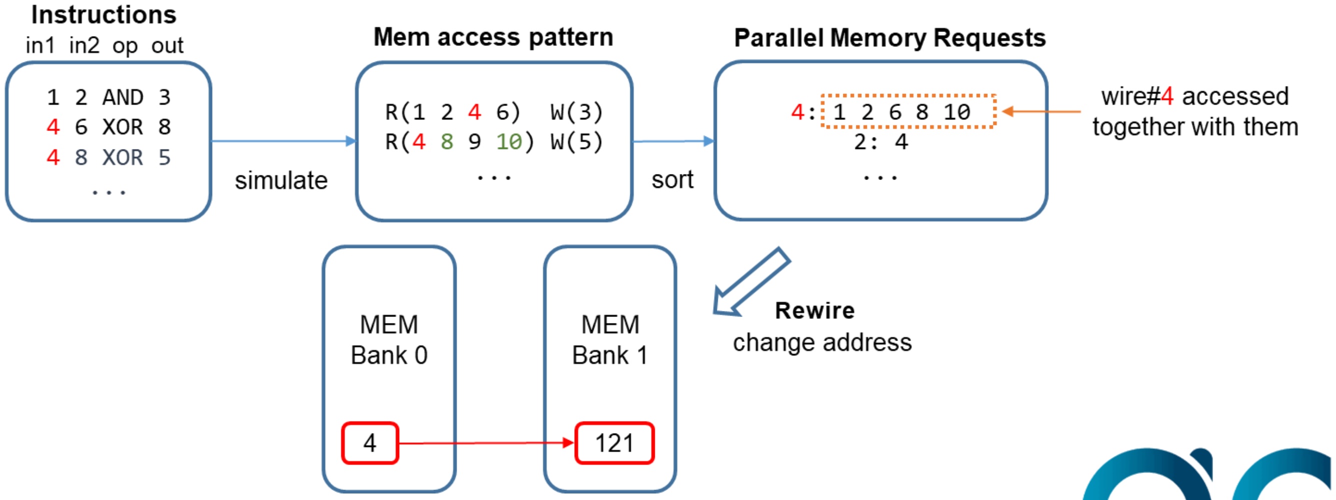
f (a, b):
    return (a>b)

```
1 2 AND 3
3 4 XOR 5
  ...
```

```
1 2 AND 3
3 4.XOR 5
```

- Resolve Memory Bank Conflicts





```
1 2 AND 3
4 6 XOR 8
4 8 XOR 5
   ...
```

```
R(1 2 4 6)  W(3)
R(4 8 9 10) W(5)
     ...
```

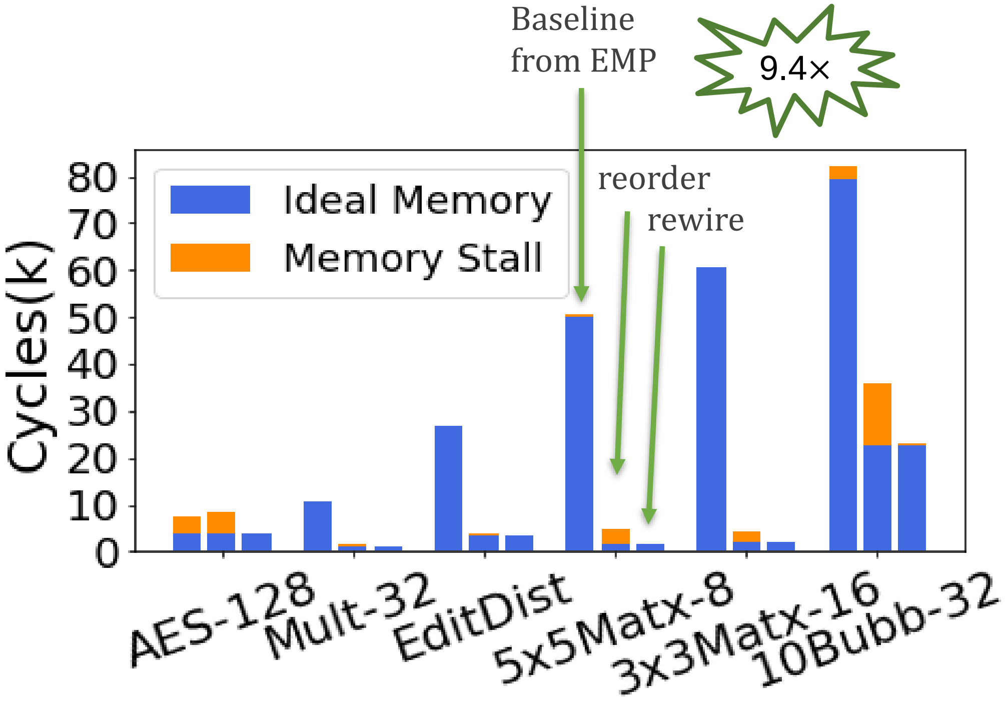
```
4: 1 2 6 8 10
   2: 4
    ...
```

Applications Driving Architectures

Performance

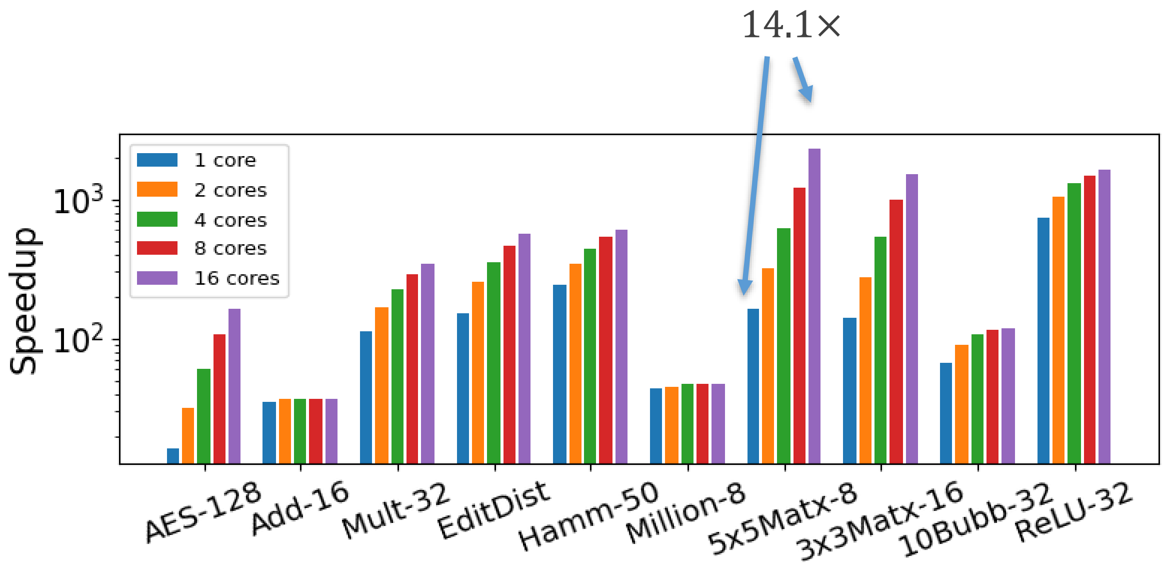
Optimized Compiler

- Reorder: 2.3× overall speedup (geomean), but 5.5× more memory stalls
- Reorder + Rewire: 4.04× overall speedup



Multi-Core Scaling (1, 2, 4, 8, 16)

- Overall 1→16 cores speedup: 2.76×
- Comparing with software: overall 97.8× speedup (only a single GCcore), 258× speedup (16-core)



Applications Driving Architectures

How HAAC achieves ADA goals

HAAC Goals

- Wide-scale deployment of practical privacy-preserving computation
- Cryptographically secure

Aligns with ADA Task 2.7 – Privacy-Enhanced Computation

- Hardware-software co-designed GC accelerator achieving an average speedup of 258×

JUMP

Joint University Microelectronics Program

www.src.org/program/jump

Semiconductor Research Corporation

@srcJUMP