BMBS
Bare Metal Benchmark Suite

Alexander Wachter, IAIK
https://github.com/alexanderwachter/bmbs

18. Oktober 2017

What is BMBS

A benchmark suite for microcontrollers

- A benchmark
- On bare metal (no OS)
- A suite

Why do we need BMBS?

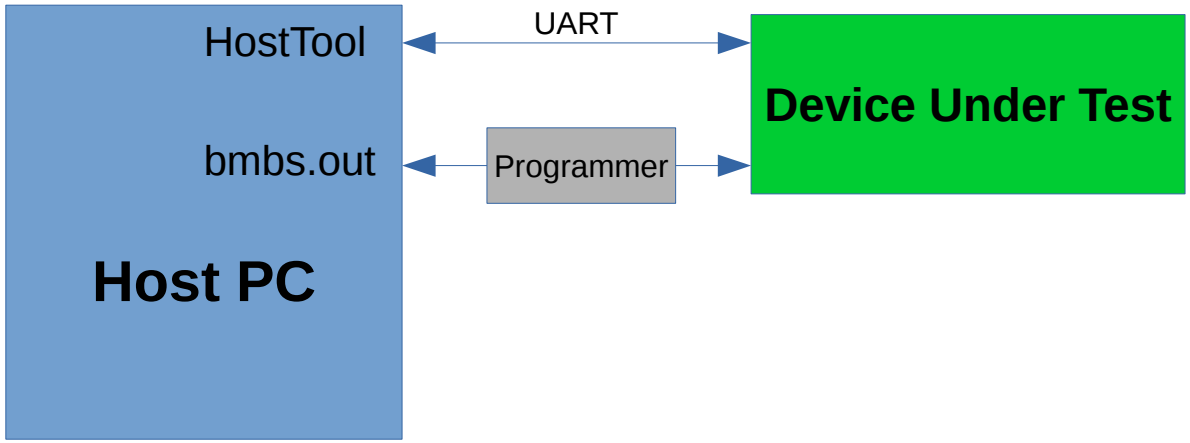
Today's benchmarks

- Use OS support
- Need a lot of resources
- Made for high performance CPUs

For microcontrollers it is necessary to have

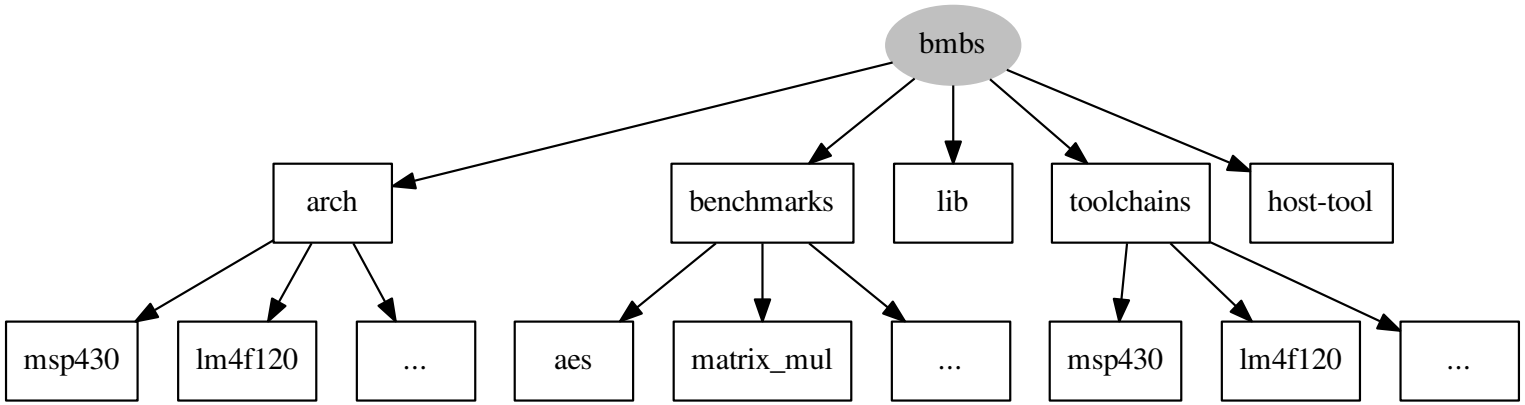
- Platform and OS independence
- Communication
- Cross compiling toolchain
- (Open Source license)

General Setup

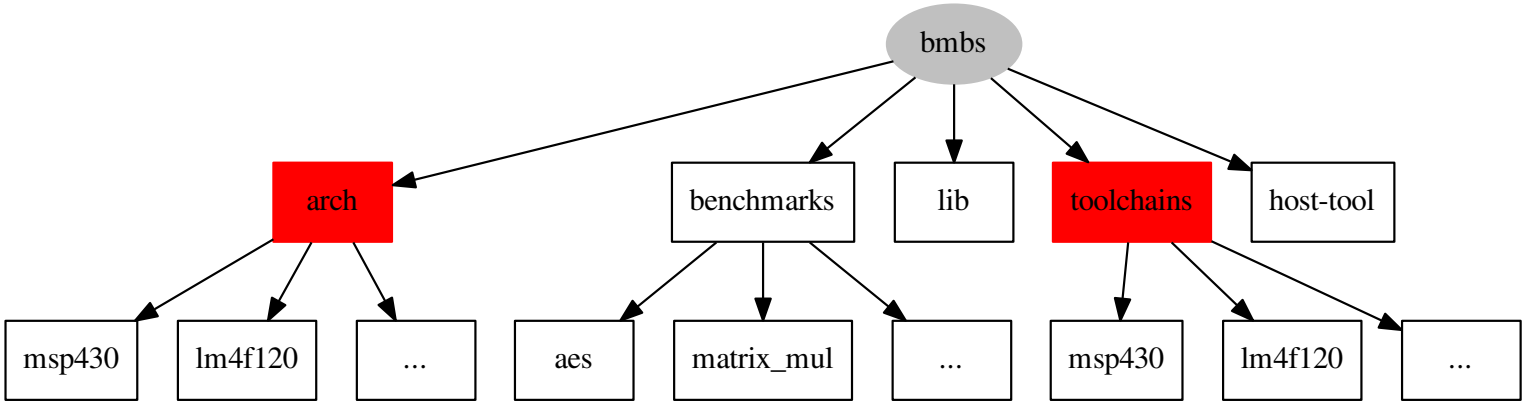


- Running host-tool
- Cross-compile
- Program
- Run benchmarks
- Return results

Structure

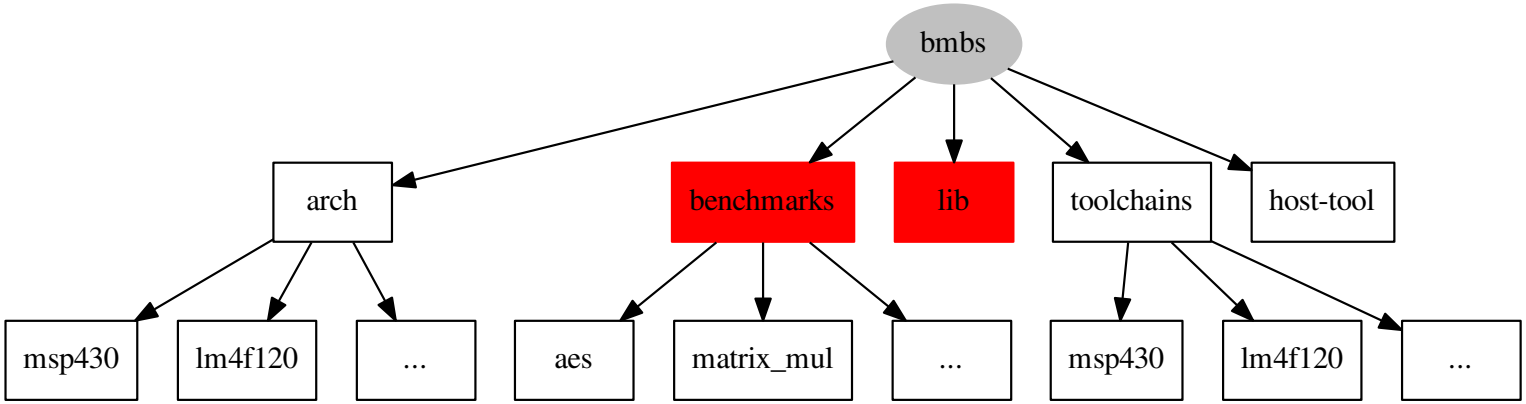

bmbs
arch
benchmarks
lib
toolchains
host-tool
msp430
lm4f120
...
aes
matrix_mul
...
msp430
lm4f120
...

Platform depending code



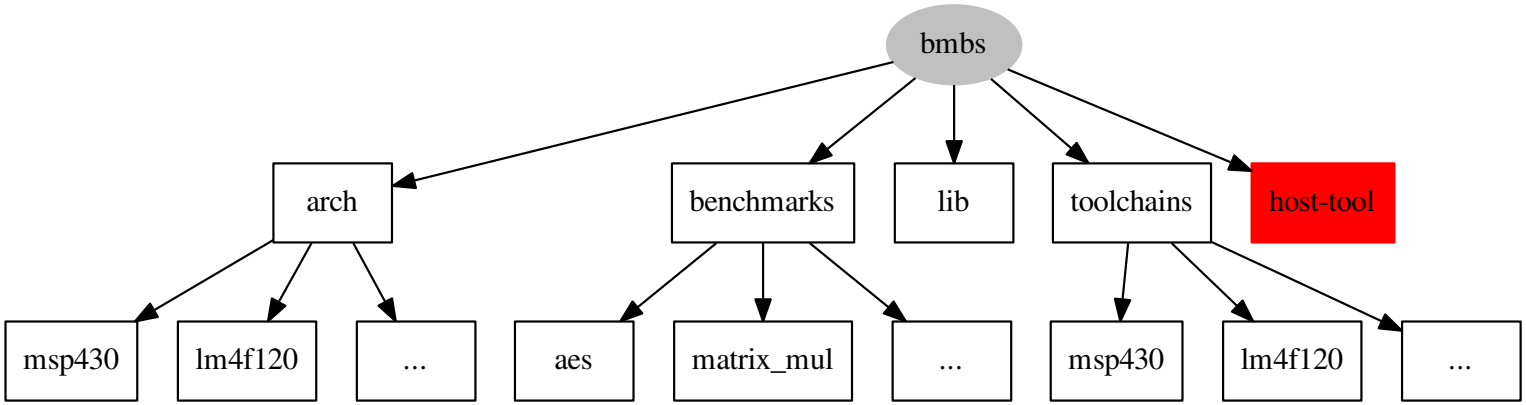
- Initialization of clocks and watchdog
- UART with interrupt
- System time
- Reset

Platform independent code



- Time measurement
- printf
- Result feedback
- Registration
- GPIO
- String-lib
- Benchmark code

Host-Tool


bmbs
arch
benchmarks
lib
toolchains
host-tool
msp430
lm4f120
...
aes
matrix_mul
...
msp430
lm4f120
...

Evaluation

- Benchmarks
  - Matrix Multiplication
  - List Operations
  - AES
- Platforms
  - ATmega328p
  - MSP430F5529
  - LM4F120

Matrix Multiplication

- Multiplication of [10*x*5] · [5*x*10]
- Two implementations
  - Based on array indexes
  - Based on pointer arithmetic
- 8 bit, 16 bit and 32 bit values

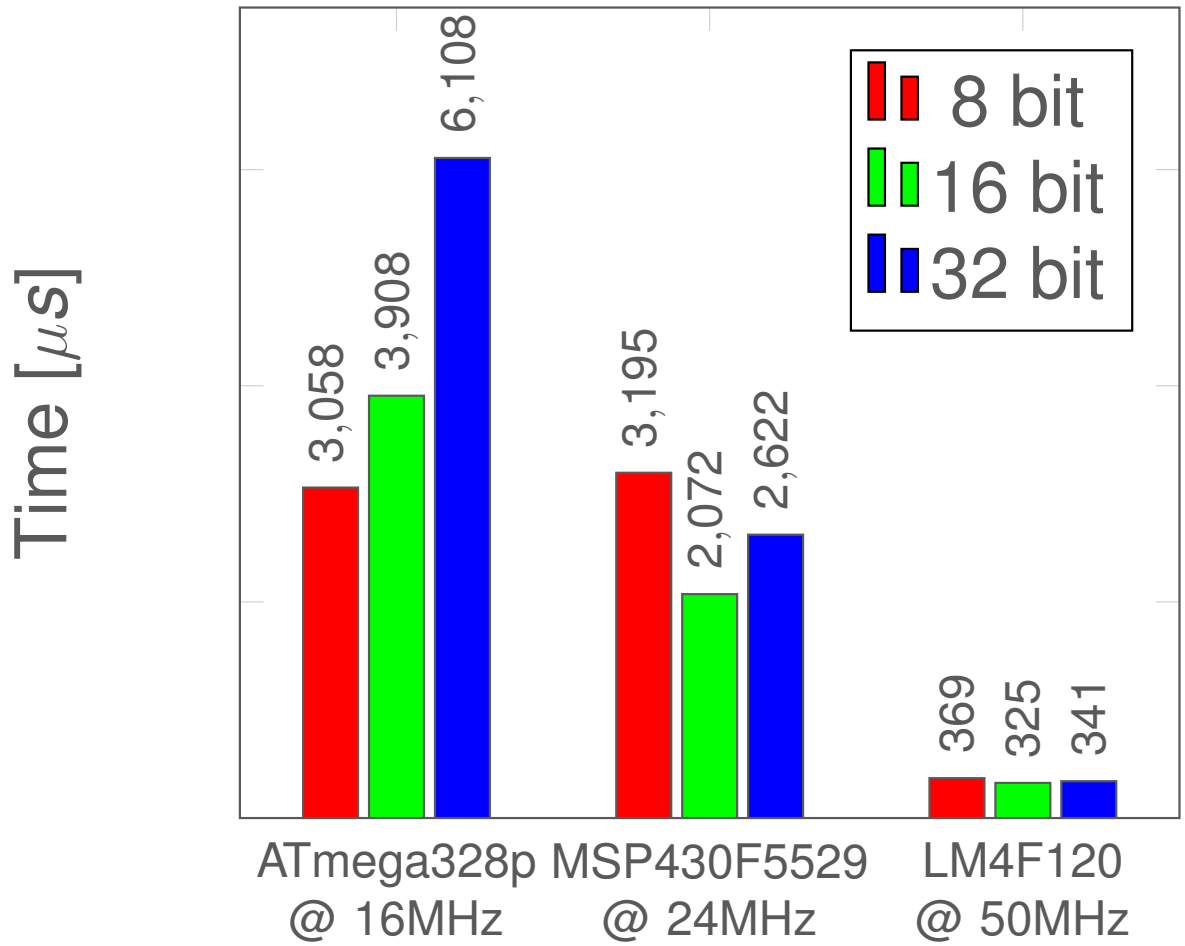
Matrix Multiplication Index Based

```
for(i = 0; i < res->m; i++)
 {
   for(j = 0; j < res->n; j++)
   {
     for(k = 0; k < a->n ;k++)
       res->data[i * res->n + j] +=
             a->data[i * a->n + k]
           * b->data[k * b->n + j];
   }
 }
```

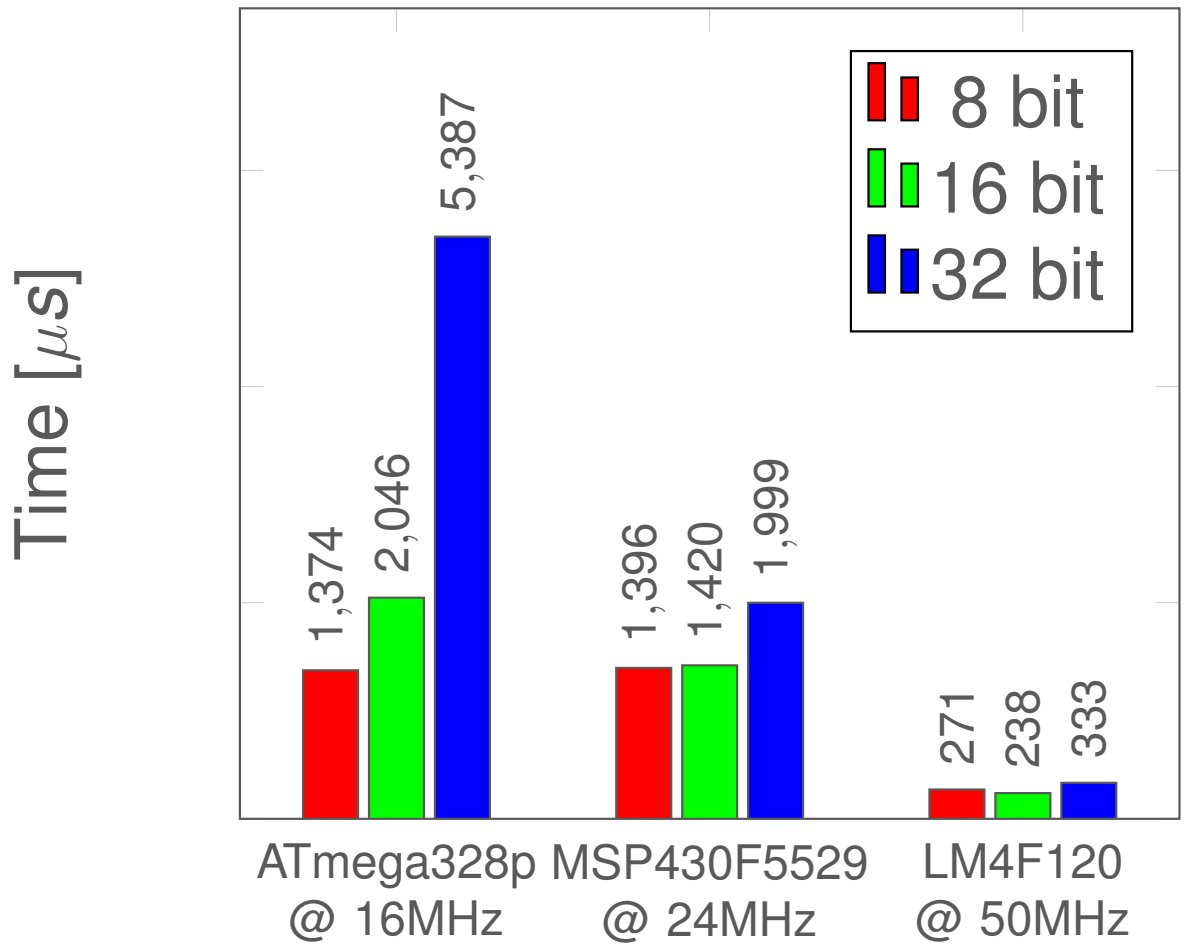
Matrix Multiplication Pointer Based

```
for(i = 0, a_data_save = a->data;
    i < res->m;
    i++, a_data_save += a->n)
{
  for(j = 0; j < res->n; j++, res_data++)
  {
    b_data = b->data + j;
    for(k = 0, a_data = a_data_save;
        k < a->n;
        k++, b_data += b->n, a_data++)
      *res_data += (*a_data) * (*b_data);
  }
}
```

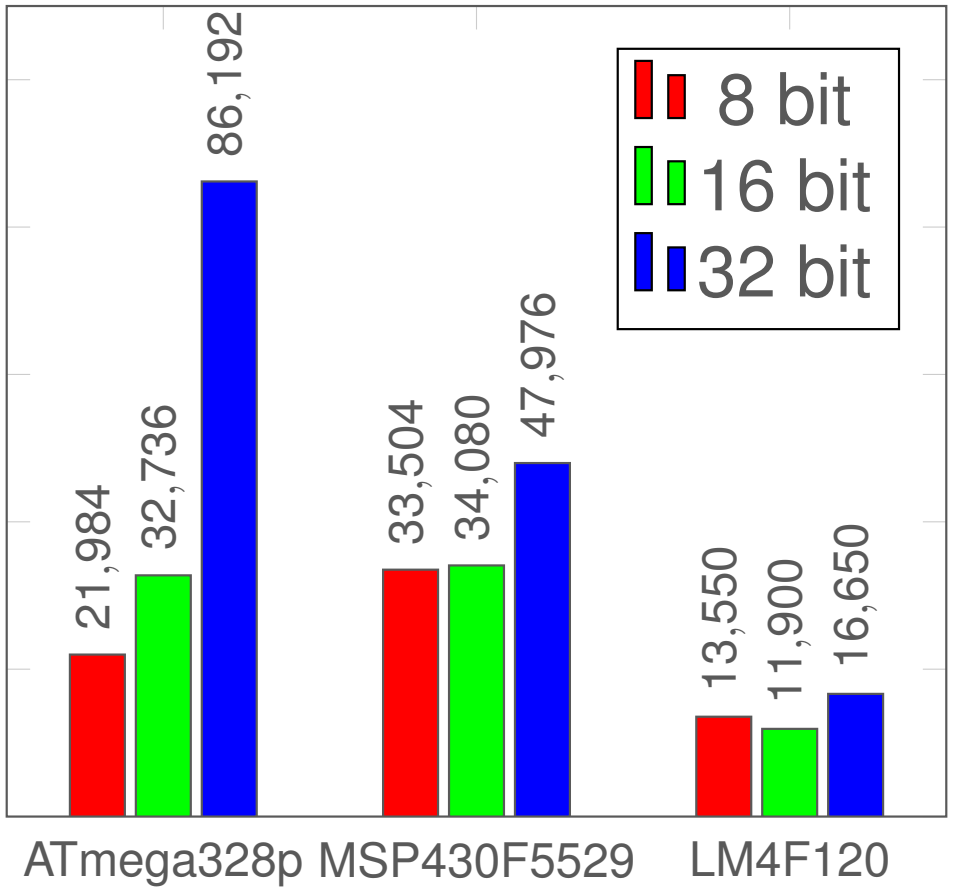
Index Based Matrix Multiplication Results



Pointer Arithmetic Based Matrix Multiplication Results



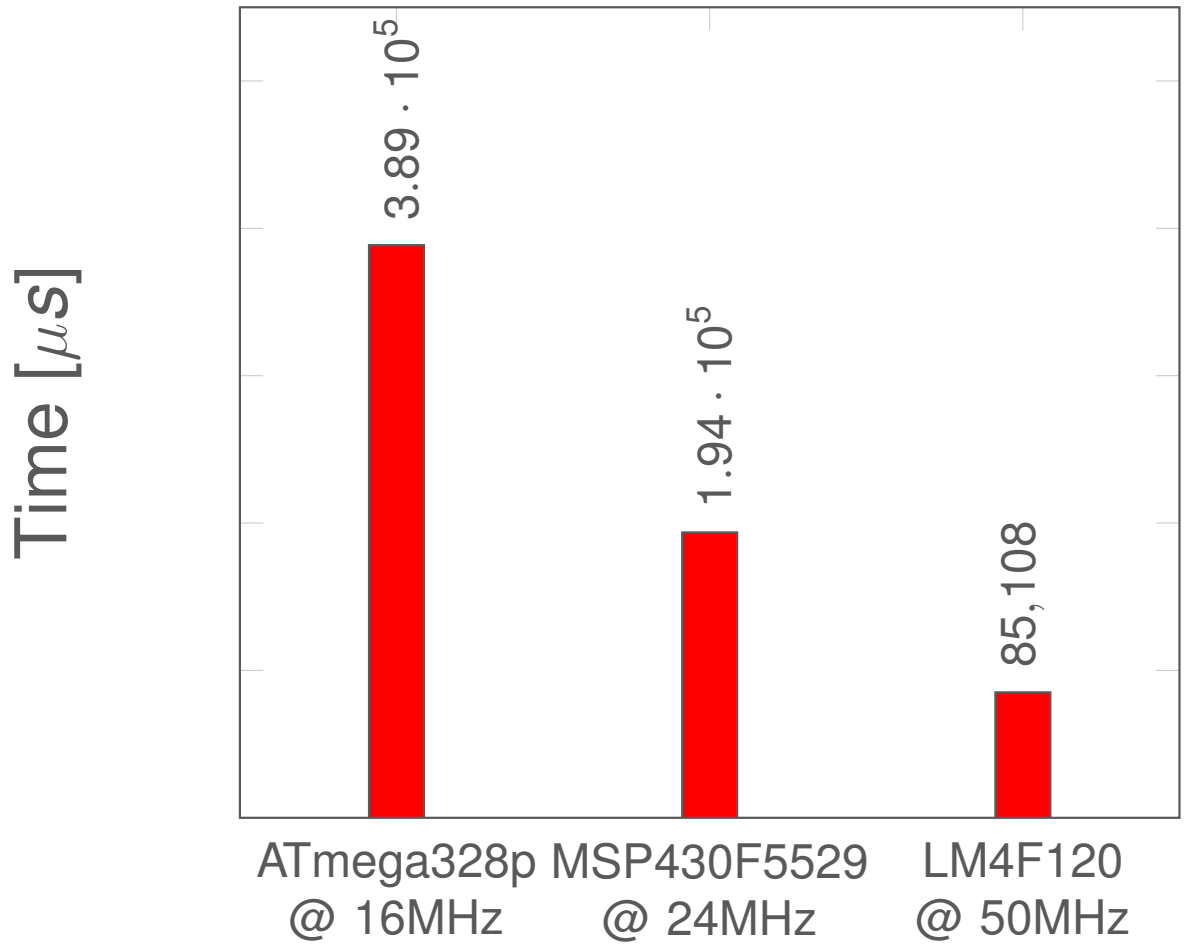
Pointer Arithmetic Based Matrix Multiplication Results Normalized by Core Frequency



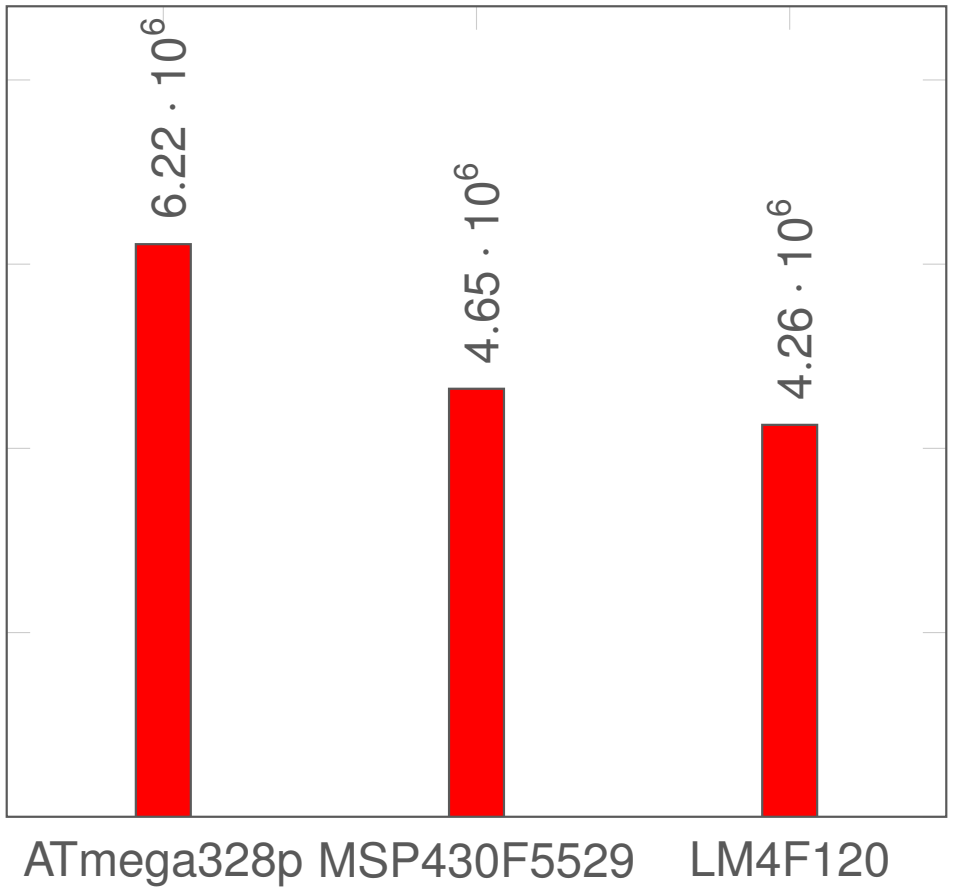
List Operations

- Basic linked list with pointer to next and pointer to data
- Operations:
  - Insert
  - Replace
  - Delete
  - Sort

List Operations Results



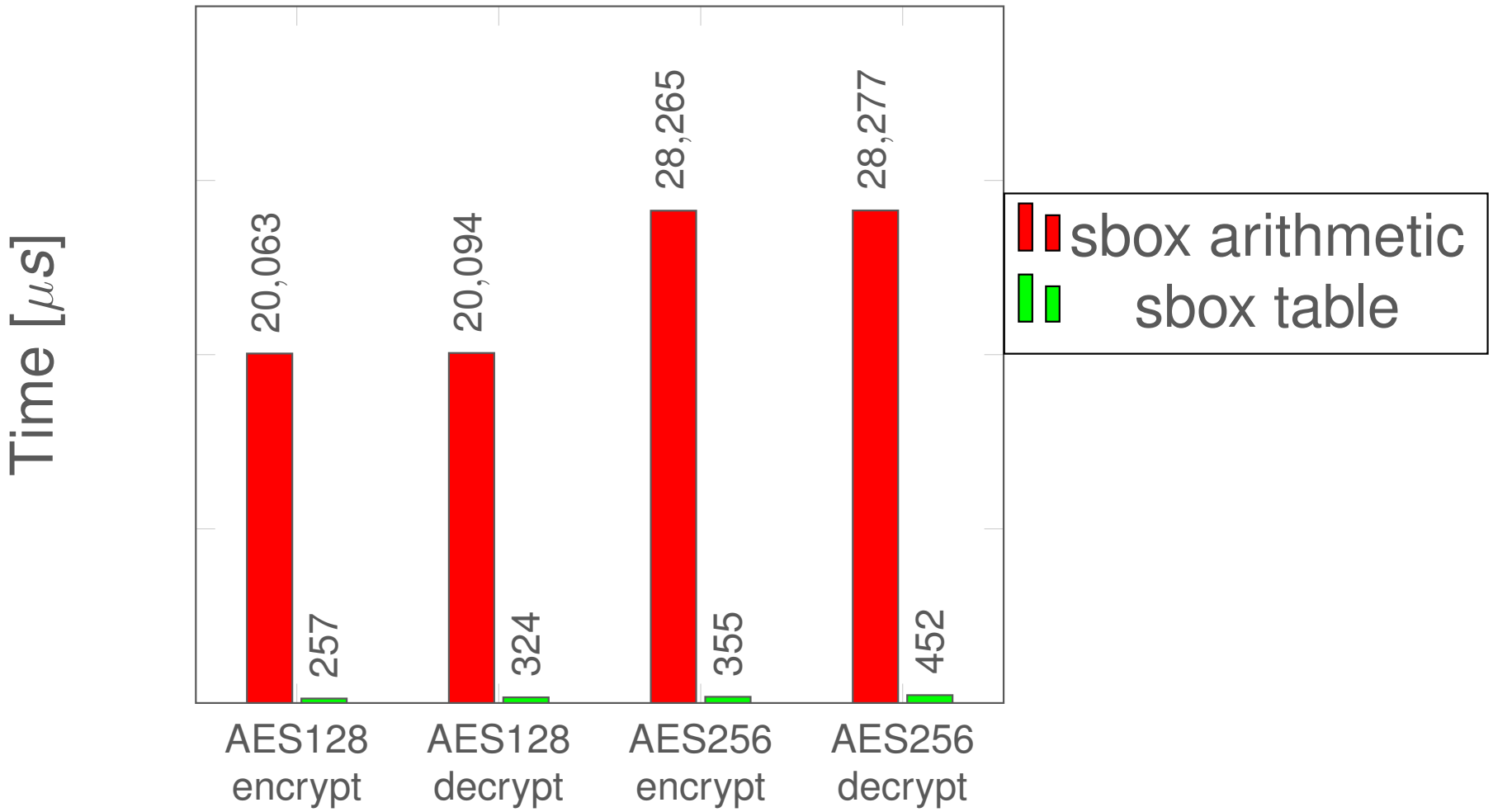
List Operations Results Normalized by Core Frequency



AES

- AES128 with arithmetic sbox
- AES256 with arithmetic sbox
- AES128 with table sbox
- AES256 with table sbox

AES Results on LM4F120



Implement your own benchmark

- Create subdirectory
- Create C file
- Implement algorithm and place time measurements
- Report results
- Register benchmark

```
void benchmark_foo(void)
{
  time_t measurement_1;
  xprintf("Hello World\n");

  tick(&measurement_1);
  ...
  time_t passed_time = tock(measurement_1);
  put_result(1, passed_time);
}

BMBS_REGISTER_BENCHMARK(Foomark, 1, 0,
   benchmark_foo)
```

Conclusion

- Designed for microcontrollers
- Various platforms
- Easy to test own algorithms
- MIT License

Thanks for your attention

https://github.com/alexanderwachter/bmbs