Advanced Micro-controllers

Agenda

- Revision - ARM Cortex-M Architecture
- ARM Assembly Language
  - Iteration/Loop
  - Barrel shifter: Instructions and Inline
  - Global variables
  - Arrays
  - Stack

ARM Assembly Language

- Loop:

```
    mov r7, #1      @ r7 = 1
loop:
    cmp r7, r2      @ while(r7 <= r2) {
    bgt loop_end
    // loop body ...
    add r7, #1      @   r7++
    b loop
loop_end:           @ }
```

- Collatz Conjure -- Math series

```
while(n >= 1) {
    if(n % 2 != 0)
        n = 3 * n + 1;  // odd * 3 + 1
```

```
    else
        n = n / 2;      // even / 2
}
```

```
while(n >= 1) {
    if(n & 1)
        n = n << 1 + n + 1;  // odd * 3 + 1
    else
        n = n >> 1;      // even / 2
}
```

- Shift operations
  - Example

```
int x = ___;
z = x >> 1;   // All bits shift to right by 1,
              // LSB discarded and MSB is copied for new MSB
unsigned int y = ___;
w = y >> 1;   // All bits shift to right by 1,
              // LSB discarded and 0 is copied for new MSB
```

  - ARM LSR instruction i.e. Logical Shift Right -- Same as Right shift on Unsigned num.
    - All bits shift to right by 1, LSB discarded and 0 is copied for new MSB
    - Equivalent to Divide by Power of 2.
  - ARM ASR instruction i.e. Arithmetic Shift Right -- Same as Right shift on Signed num.
    - All bits shift to right by 1, LSB discarded and MSB is copied for new MSB
  - Example

```
y = x << 1;
```

    - ARM LSL instruction i.e. Logical Shift Left -- Same as Left Shift operator
        - All bits shift to left by 1, MSB discarded and 0 added at LSB.
- Load and Store Multiple
    - Refer screenshot and slides.
- Stack types
    - Full Descending -- ARM Cortex-A, ARM Cortex-M
    - Empty Descending -- ARM Cortex-A,
    - Full Ascedning -- ARM Cortex-A,
    - Empty Descending -- ARM Cortex-A,
    - Refer slides