



Computer architecture

ARM assembly

ARM

<https://cpulator.01xz.net/>

Choose a system to simulate

Architecture

Any
Nios II
ARMv7
MIPS32r5
MIPS32r5 (no delay slots)
MIPS32r6
MIPS32r6 (no delay slots)

System

ARMv7 generic
ARMv7 DE1-SoC
ARMv7 DE1-SoC (v16.1)
Nios II generic
Nios II DE1-SoC
Nios II DE1-SoC (v16.1)
Nios II DE2-115

<https://cpulator.01xz.net/?sys=arm-de1soc>

Go

Demo examples – blinking led:

```
.global _start
_start:
    ldr r0,=LED
    MOV r2,#0b00000001

_write:
    str r2,[r0]
    mov r5,#0
_delay:
    add r5,r5,#1
    cmp r5,#102400
    bne _delay
    str r3,[r0]
    mov r5,#0
_delay2:
    add r5,r5,#1
    cmp r5,#102400
    bne _delay2
    str r2,[r0]
    b _write

.data
.equ LED, 0xFF200000
```

Demo examples - counter

```
.global _start
_start:
    ldr r0,=LED
    MOV r2,#0b0000001

_write:
    str r2,[r0]
    mov r5,#0
_delay:
    add r5,r5,#1
    cmp r5,#102400
    bne _delay
    mov r5,#0
_delay2:
    add r5,r5,#1
    cmp r5,#102400
    bne _delay2
    str r3,[r0]
    add r2,r2,#1
    str r2,[r0]
    b _write

.data
.equ LED, 0xFF200000
```

Demo examples – rolling led

```
.global _start
_start:
    ldr r0,=LED
    MOV r2,#0b0000001

_write:
    str r2,[r0]
    mov r5,#0
_delay:
    add r5,r5,#1
    cmp r5,#102400
    bne _delay
    mov r5,#0
_delay2:
    add r5,r5,#1
    cmp r5,#102400
    bne _delay2
    str r3,[r0]
    ROR r2,#1
    str r2,[r0]
    b _write

.data
.equ LED, 0xFF200000
```

Demo examples – control:

```
.global _start
_start:
    ldr r0,=LED
    MOV r2,#0b00000001
    ldr r1,=switch
    mov r3,#0

_write:
    str r2,[r0]
    mov r5,#0
_delay:
    add r5,r5,#1
    cmp r5,#102400
    bne _delay
    str r3,[r0]
    str r2,[r0]
    ROR r2,#1
_wait:
    ldr r6,[r1]
    cmp r6,#1
    beq _wait
    b _write

.data
.equ LED, 0xFF200000
.equ switch, 0xff200040
```

Demo examples UART:

```
.global _start
_start:
    ldr r0,=UART
    ldr r1,=TEXT_

_write:
    ldrb r2,[r1]
    cmp r2,#0
    beq _stop
    str r2,[r0]
    add r1,r1,#1
    b _write

_stop:
    b _stop

.data
.equ UART, 0xFF201000
TEXT_:
.asciz "Hello world from ARM!"
.end
```


Tasks:

- Output numbers 1,2,3,4,5,6 to seven segment leds (left to right)
- Count from 1-9 on seven segment display
- Make leds move from left to right WHEN the switch is ON.



**Thank you for
your attention!**