

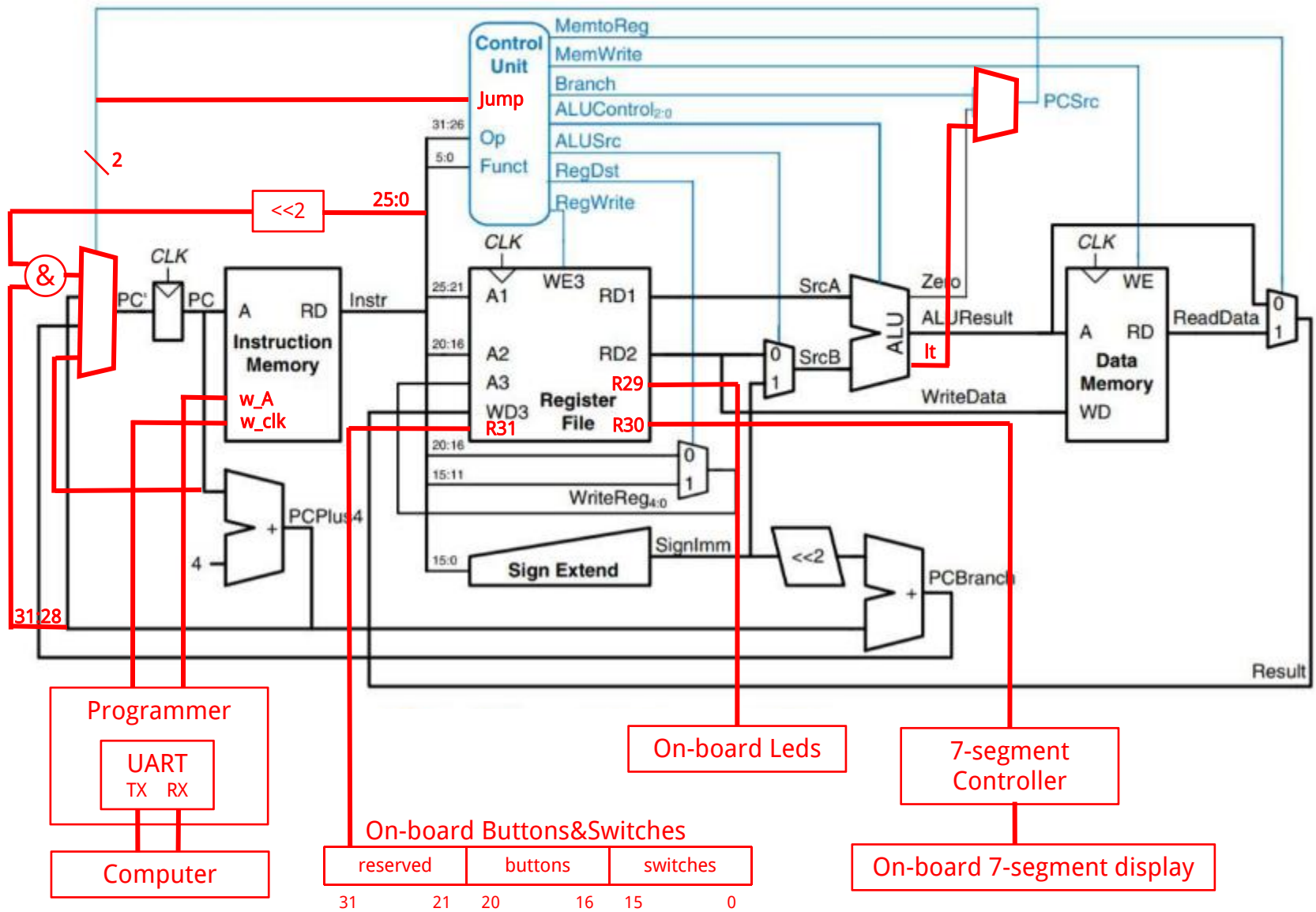
RC5 Assembly to Binary

```
350 NOR r10 r10 r0    # r10 <- A^B = (B nand (A nand 1))
351 OR r11 r10 r10    # r11 <- r10
352
353 ORI r6 r0 0        # rotation counter
354 AND r16 r26 r8     # A: 5 LSB rotation bits
355 BEQ r0 r0 2
356 SHL r11 r11 1      # r11 <- r11 << 1
357 ADDI r6 r6 1        # r6 += 1
358 BLT r16 r6 -3      # loop if r6 < rotation bits
359
360 ORI r16 r0 32       # total rotation bits = 32
361 BEQ r0 r0 2
362 SHR r10 r10 1      # r10 <- r10 >> 1
363 ADDI r6 r6 1        # r6 += 1
364 BLT r16 r6 -3      # loop if r6 < 32
365
366 OR r10 r11 r10     # r10 <- rotr(A^B, A)
```

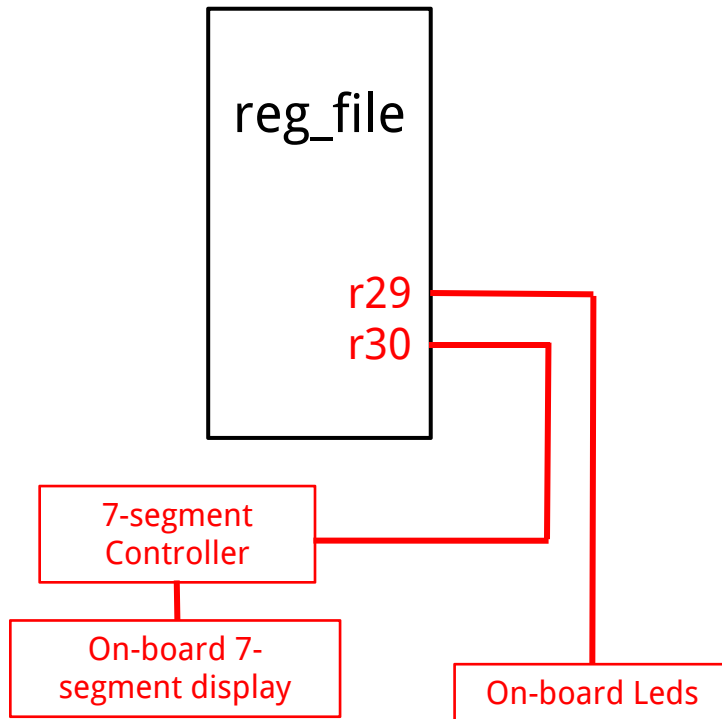
Part of RC5 assembly
(rotate operation)

```
decode = {
  'ADD': {'type': 'R', 'op': 0x00, 'func': 0x10},
  'ADDI': {'type': 'I', 'op': 0x01},
  'SUB': {'type': 'R', 'op': 0x00, 'func': 0x11},
  'SUBI': {'type': 'I', 'op': 0x02},
  'AND': {'type': 'R', 'op': 0x00, 'func': 0x12},
  'ANDI': {'type': 'I', 'op': 0x03},
  'OR': {'type': 'R', 'op': 0x00, 'func': 0x13},
  'NOR': {'type': 'R', 'op': 0x00, 'func': 0x14},
  'ORI': {'type': 'I', 'op': 0x04},
  'SHL': {'type': 'I', 'op': 0x05},
  'SHR': {'type': 'I', 'op': 0x06},
  'LW': {'type': 'I', 'op': 0x07},
  'SW': {'type': 'I', 'op': 0x08},
  'BLT': {'type': 'I', 'op': 0x09},
  'BEQ': {'type': 'I', 'op': 0x0a},
  'BNE': {'type': 'I', 'op': 0x0b},
  'JMP': {'type': 'J', 'op': 0x0c},
  'HAL': {'type': 'J', 'op': 0x0f}, }
```

Decoding table



Leds and 7-segment display



Example code:

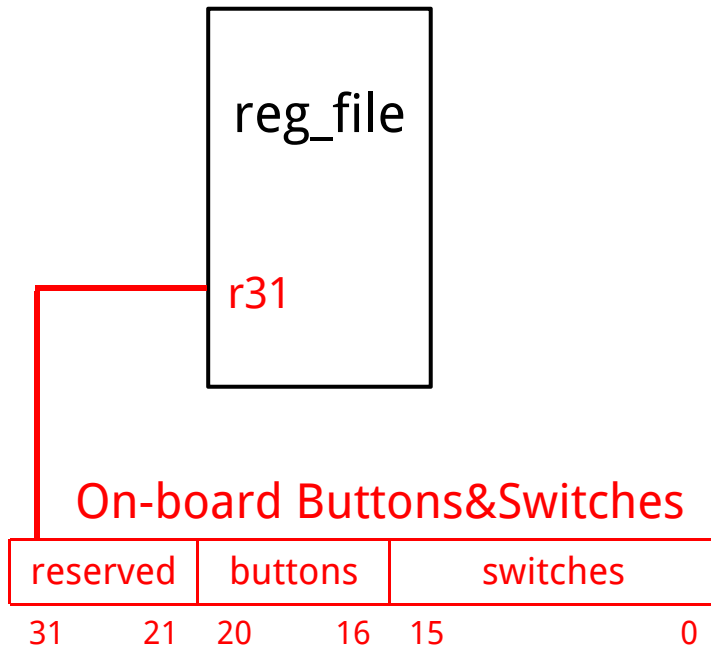
```
# Turn on led(7)
# r0 = 0, r1 = 1
SHL r28, r1, 7      # r28(7) = 1
OR  r29, r29, r28
```

```
# display data[30]
LW  r30 r0 30
```

VHD pseudocode:

```
led <= r29(15 downto 0);
U_7seg_ctrl(r30, seg, an);
```

Buttons & Switches



VHD pseudocode:

```
r31(20 downto 0) <= btn&sw;
```

Example code:

```
# Wait for btn0
```

```
# r22 = 1<<16 for picking up r31(16)
```

```
LOOP:
```

```
AND r28, r22, r31    # pick r31(16)
```

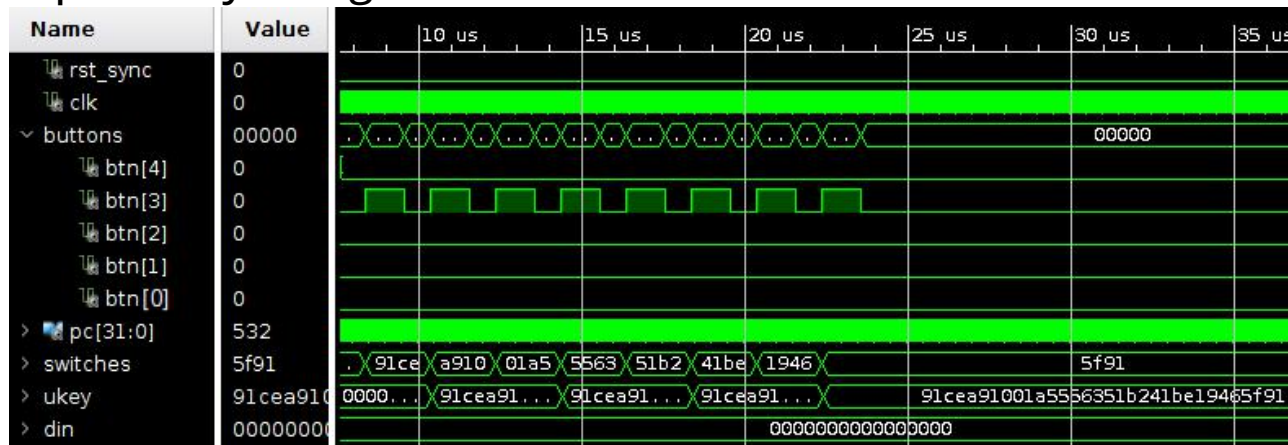
```
BNE r28, r22, LOOP
```

```
# Do something here...
```

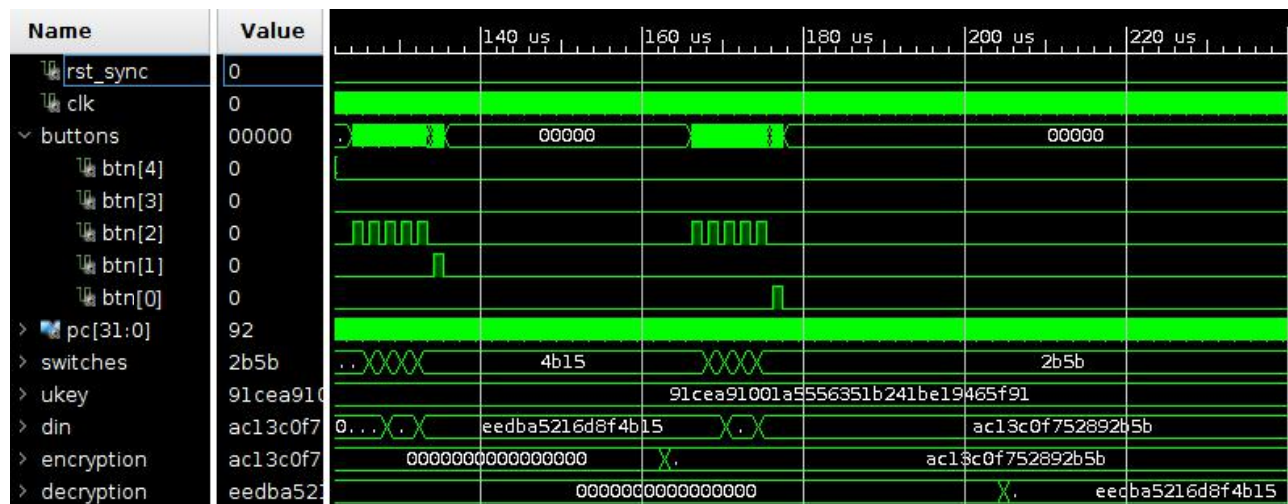
- If btn0 is not pressed, r31 = '0'. BNE becomes True. The code will keep looping.
- If btn0 is pressed, r31 = '1'. BNE becomes False. The code will continue.

Buttons & Switches

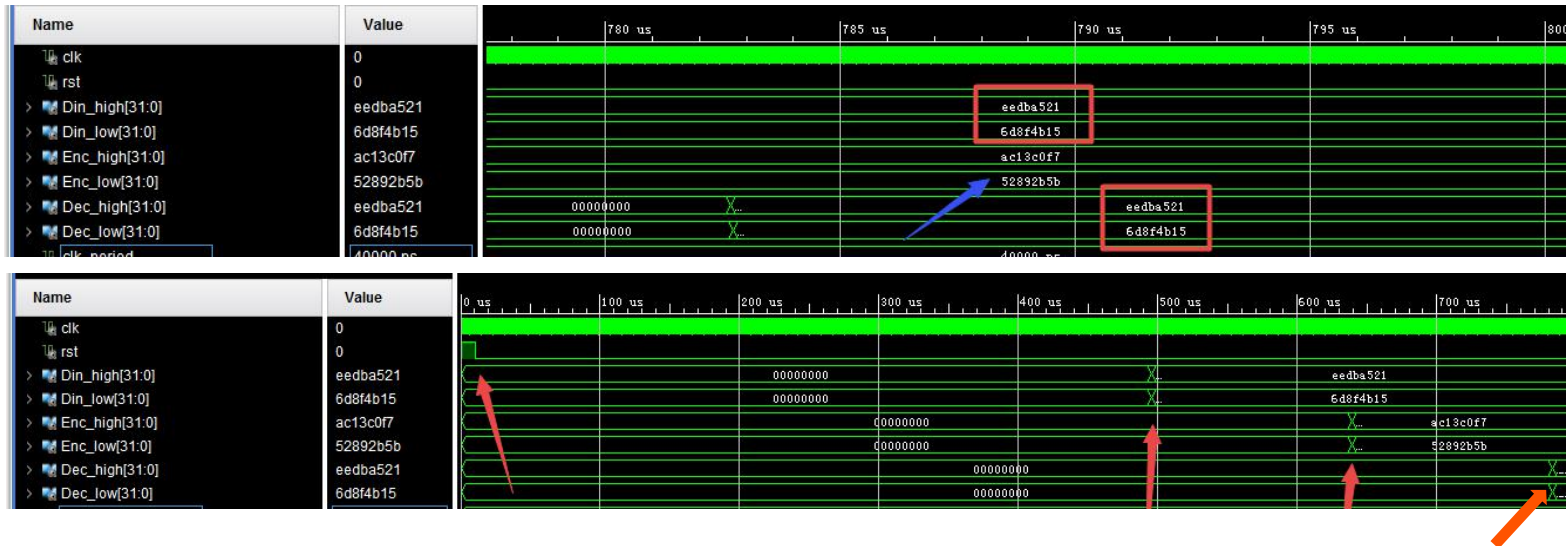
Input Ukey using btn3:



Input Din (btn3); start encryption/decryption (btn1/btn0)



RC5 Timing



Critical path delay: 38 ns

Clock frequency: 25 MHz

Latency (cycles used in key-gen, encryption and decryption):

| | Start | End | Time-Diff | Latency | clk period |
|------------|--------|--------|-----------|---------|------------|
| Key - Gen | 10460 | 495860 | 485400 | 12135 | 40 |
| Encryption | 495860 | 639260 | 143400 | 3585 | |
| Decryption | 639260 | 782700 | 143440 | 3586 | |