

PreReg0_20

Using the indirect addressing mode instruction `st` clear registers, and initialize registers `r0` to `r20` with values `0x15` to `0x01`

Hint: SRAM address `0x0000` to `0x0001F` map to registers `r0` to `r31`

| Register | Value |
|----------|-------|
| R00 | 0x00 |
| R01 | 0x00 |
| R02 | 0x00 |
| R03 | 0x00 |
| R04 | 0x00 |
| R05 | 0x00 |
| R06 | 0x00 |
| R07 | 0x00 |
| R08 | 0x00 |
| R09 | 0x00 |
| R10 | 0x00 |
| R11 | 0x00 |
| R12 | 0x00 |
| R13 | 0x00 |
| R14 | 0x00 |
| R15 | 0x00 |
| R16 | 0x00 |
| R17 | 0x00 |
| R18 | 0x00 |
| R19 | 0x00 |
| R20 | 0x00 |

```

.INCLUDE <m328pdef.inc>
Setup:
  rcall PreReg0_20
  rjmp Setup

PreReg0_20:
  clr XH      ; r27
  clr XL      ; r26
  ldi r24,21  ; clear registers r0 to r20
reg_loop:
  st X+,r24
  dec r24
  brne reg_loop
ret
  
```

| Address | Value |
|---------|---|
| 000000 | 00 |
| 00001A | 00 00 00 00 00 00 00 00 |

Figure 1 Start of assembly program to initialize registers `r0` to `r20`. All registers cleared (reset condition).

| Register | Value |
|----------|-------|
| R00 | 0x15 |
| R01 | 0x14 |
| R02 | 0x13 |
| R03 | 0x12 |
| R04 | 0x11 |
| R05 | 0x10 |
| R06 | 0x0F |
| R07 | 0x0E |
| R08 | 0x0D |
| R09 | 0x0C |
| R10 | 0x0B |
| R11 | 0x0A |
| R12 | 0x09 |
| R13 | 0x08 |
| R14 | 0x07 |
| R15 | 0x06 |
| R16 | 0x05 |
| R17 | 0x04 |
| R18 | 0x03 |
| R19 | 0x02 |
| R20 | 0x01 |
| R21 | 0x00 |

```

.INCLUDE <m328pdef.inc>
Setup:
  rcall PreReg0_20
  rjmp Setup

PreReg0_20:
  clr XH      ; r27
  clr XL      ; r26
  ldi r24,21  ; clear registers r0 to r20
reg_loop:
  st X+,r24
  dec r24
  brne reg_loop
ret
  
```

| Address | Value |
|---------|--|
| 000000 | 15 14 13 12 11 10 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00 00 00 00 00 00 00 00 00 00 |
| 00001A | 15 00 00 00 00 00 00 00 |

Figure 2 End of assembly program to initialize registers `r0` to `r20`. Registers `r0` to `r20` set to values `0x15` to `0x01`. Registers `r21` to `r31` are not modified (reset condition).