AVR®

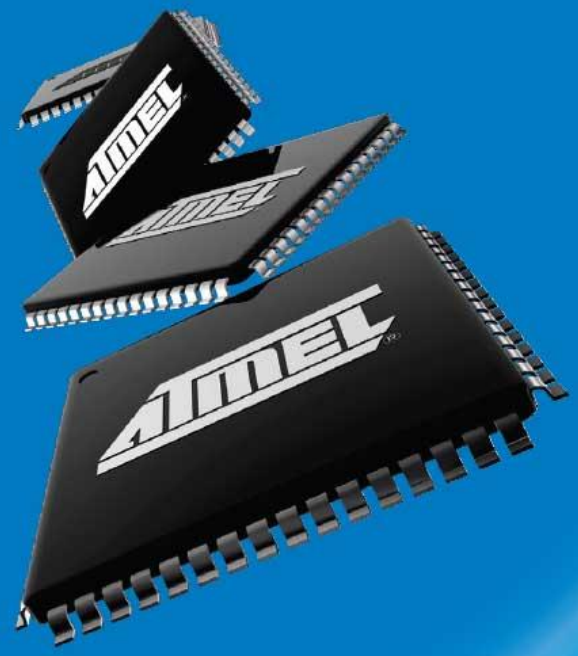
8-bit Microcontrollers

AVR®32

32-bit Microcontrollers and Application Processors

THOSE WHO ARE LAST NOW WILL BE FIRST THEN, AND THOSE WHO ARE FIRST WILL BE LAST.

MATTHEW 20:16



Introduction to AVR Assembly Language Programming II

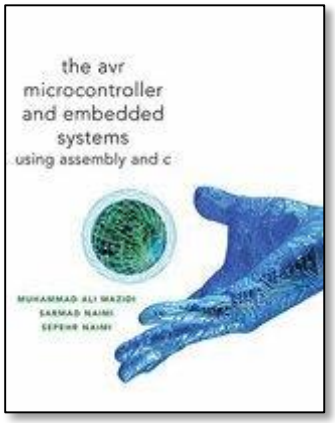
February 2009

ATMEL

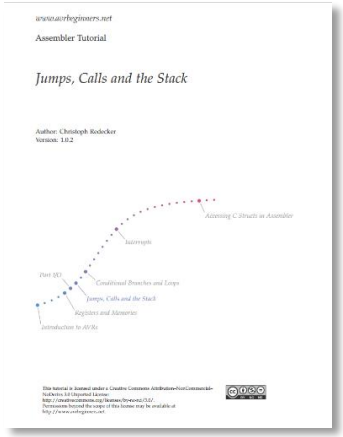
Everywhere You Are®

AVR Stack Operations

READING



The AVR Microcontroller and Embedded Systems using Assembly and C)
by Muhammad Ali Mazidi, Sarmad Naimi, and Sepehr Naimi
Chapter 3: Branch, Call, and Time Delay Loop
Section 3.2: Call Instructions and Stack



AVRBeginners.Net
Jumps, Calls and the Stack

Contents

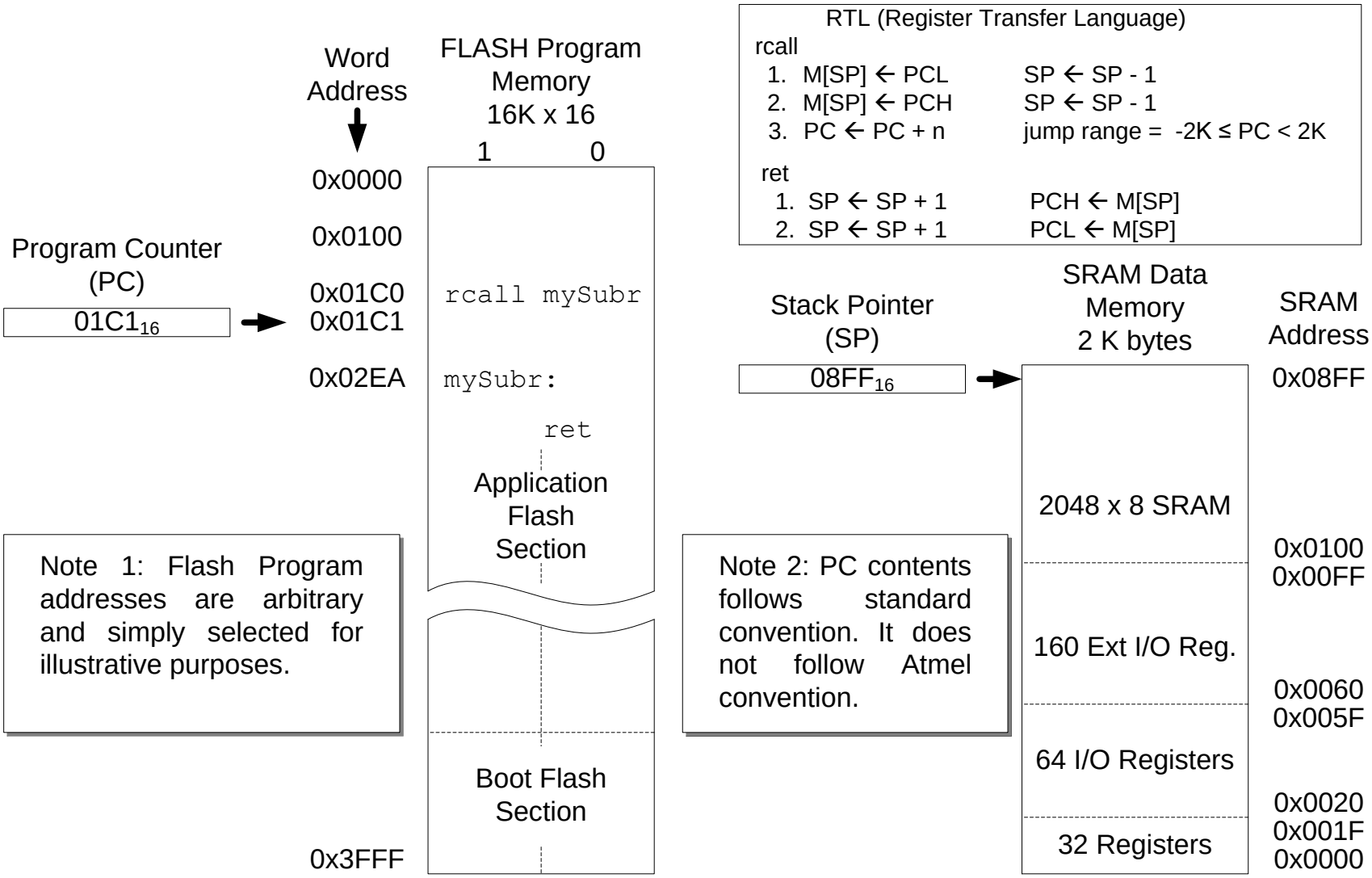
WORKING WITH STACKS

- Stacks
  - FIFO and LIFO
  - SP
  - Initialization
- LIFO Stack Operations (Push and Pop)
  - Explicit `push and pop`

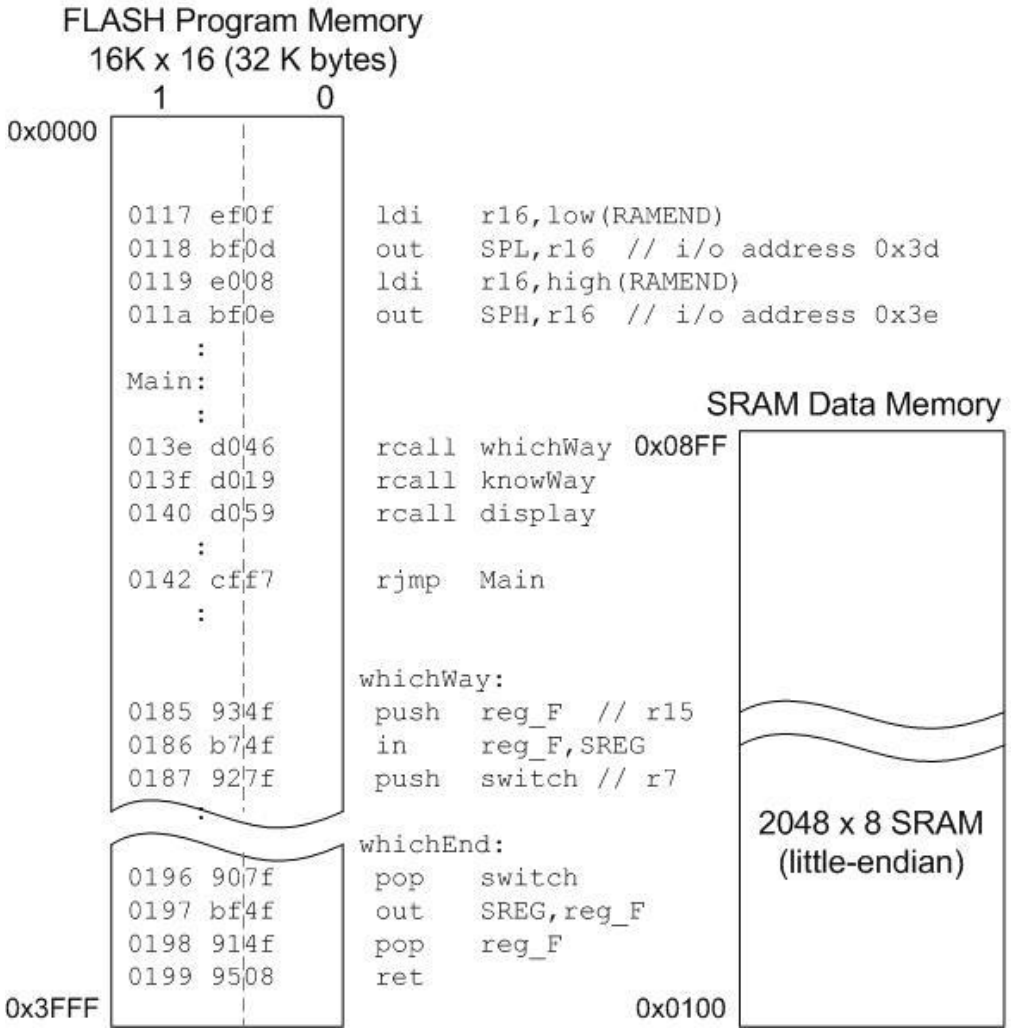
RTL (Register Transfer Language)	
push Rn	
M[SP] ← Rn	SP ← SP - 1
pop Rn	
SP ← SP + 1	Rn ← M[SP]

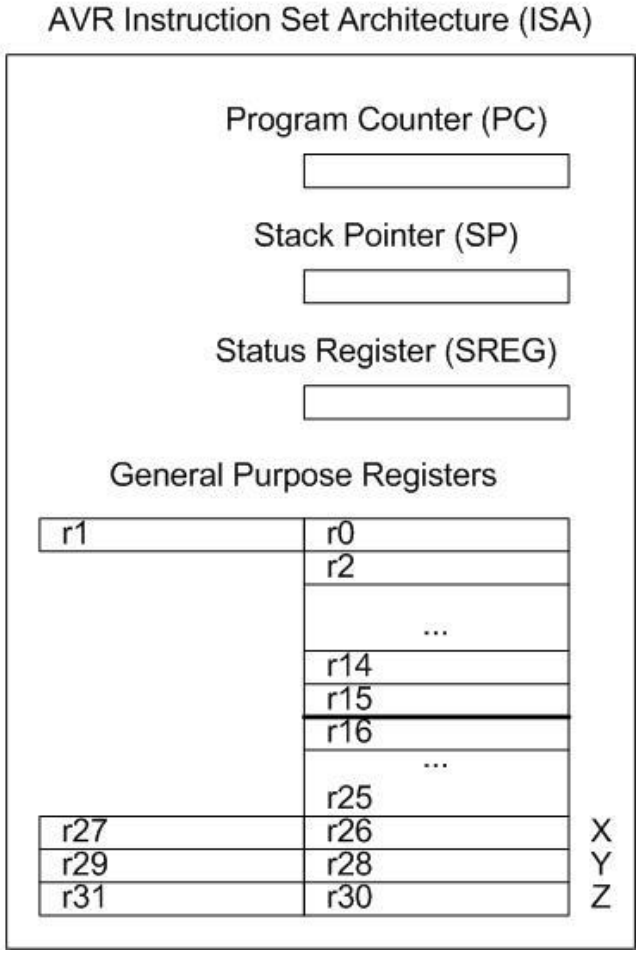
  - Implicit `rcall, call, icall, ret, reti`
- Working with the Stack (3 questions and answers)
  - Word Size 1 byte
  - Points At Empty Byte
  - Direction Decrements stack by 2 for implicit (call) and by 1 for explicit (push) stack operations
- The Program Counter byte ordering on the SRAM stack is **Big Endian**.

STACK OPERATION ON A CALL INSTRUCTION



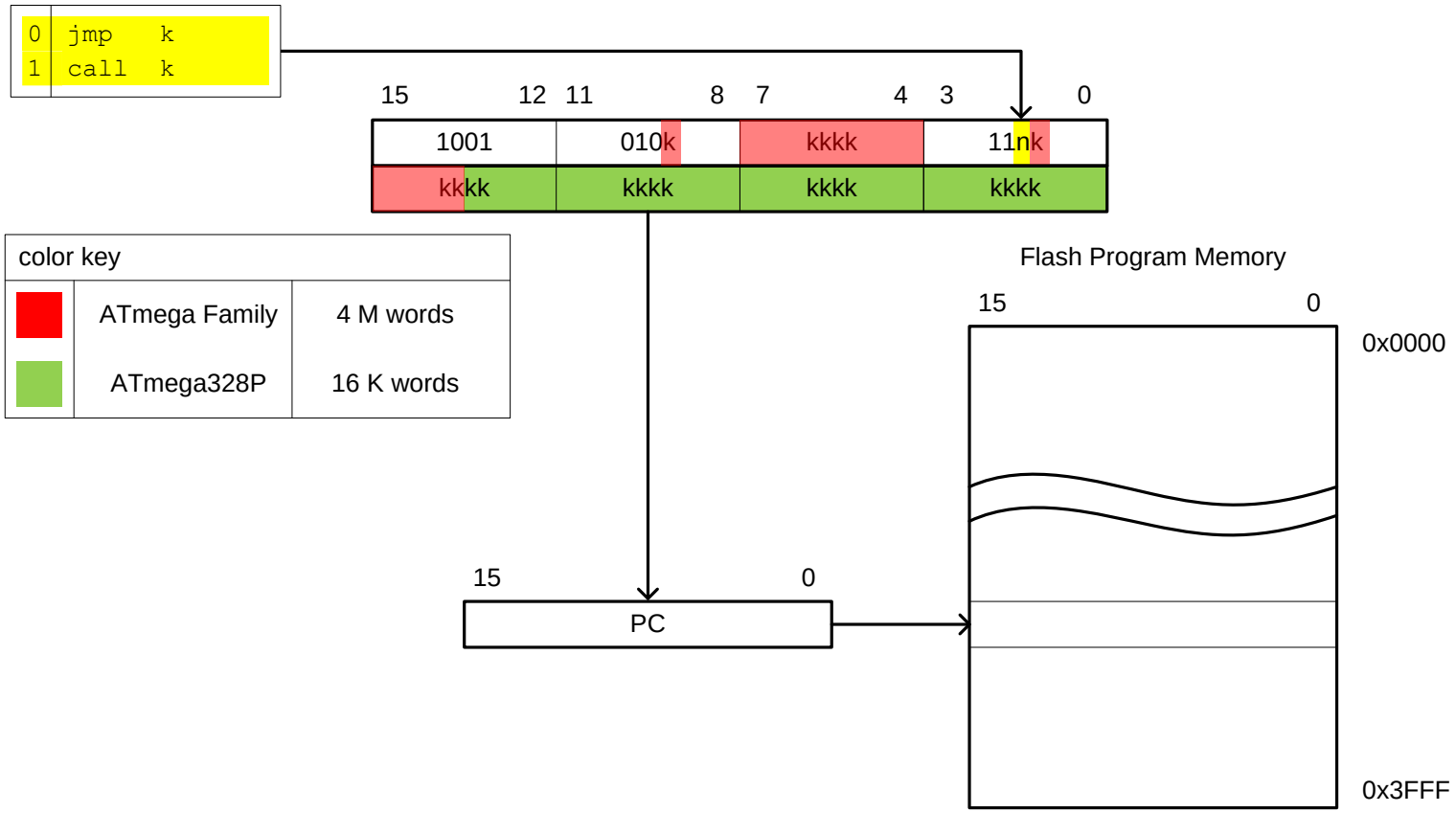
AMAZING LAB DESIGN EXAMPLE





CALL Instruction Encoding

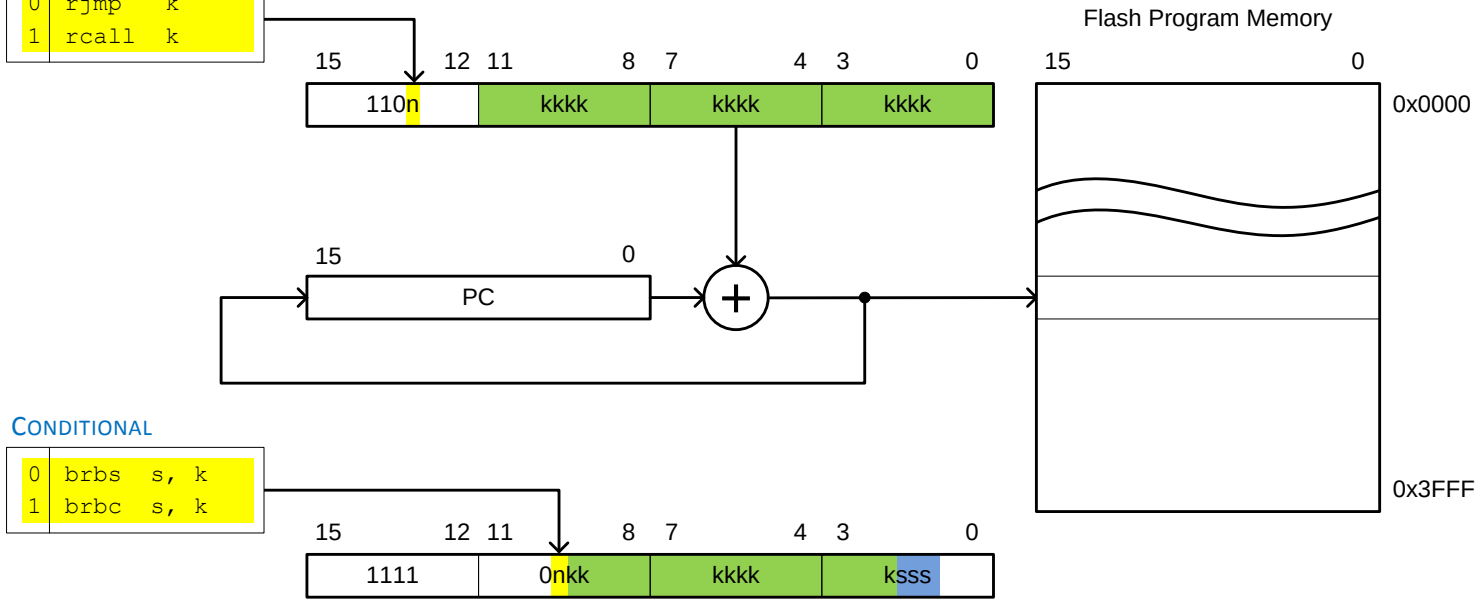
- All control transfer addressing modes modify the program counter.
- The Program Counter byte ordering on the SRAM stack is **Big Endian**.



RCALL Instruction Encoding

Unconditional

```
0  rjmp   k
1  rcall  k
```



Conditional

```
0  brbs   s, k
1  brbc   s, k
```


alias			
SREG	sss	brbs s, k	brbc s, k
I	111	brie k	brid k
T	110	brts k	brtc k
H	101	brhs k	brhc k
S	100	brlt k	brge k
V	011	brvs k	brvc k
N	010	brmi k	brpl k
Z	001	breq k	brne k
C	000	brcs k brlo k	brcc k brsh k

Notes

1. See Register Direct Addressing for encoding of skip register bit set/clear instructions sbrc and sbrs.
2. See I/O Direct Addressing for encoding of skip I/O register bit set/clear instructions sbis and sbic.

RET Instruction Encoding

nnn	opcode	notes
000		
001	ijmp	see illustration
010		
011	eijmp	n/a to 328P
100	ret	PC ← M[SP + 1]
101	icall	see illustration
110	reti	PC ← M[SP + 1]
111	eicall	n/a to 328P