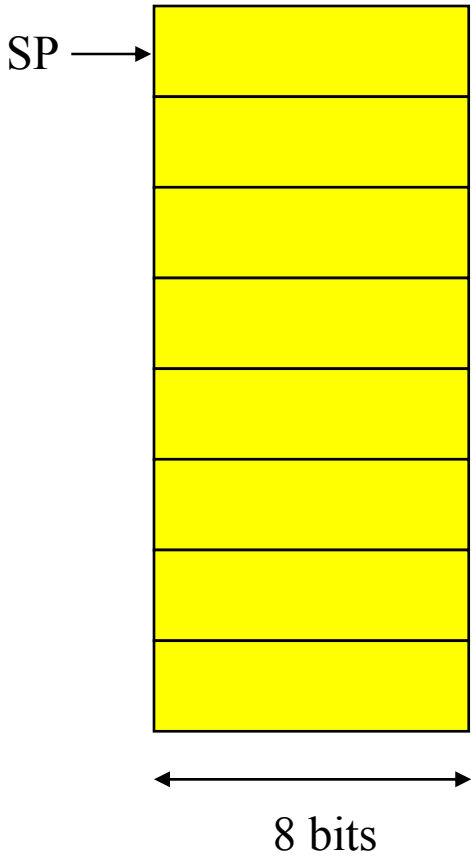
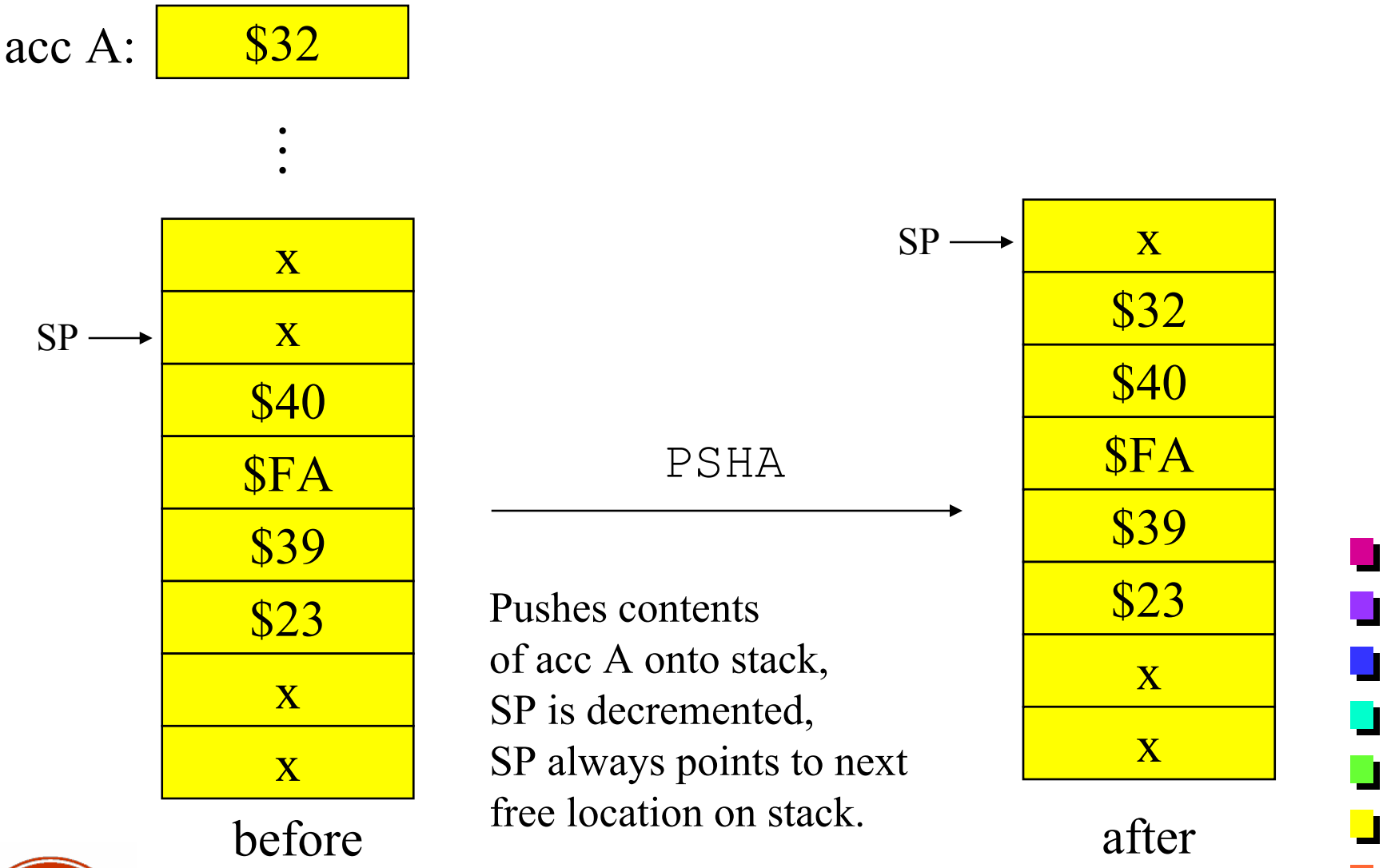
EE 5340/7340
Biomedical Instrumentation
Motorola 68HC11 Microcontroller
Lecture 2

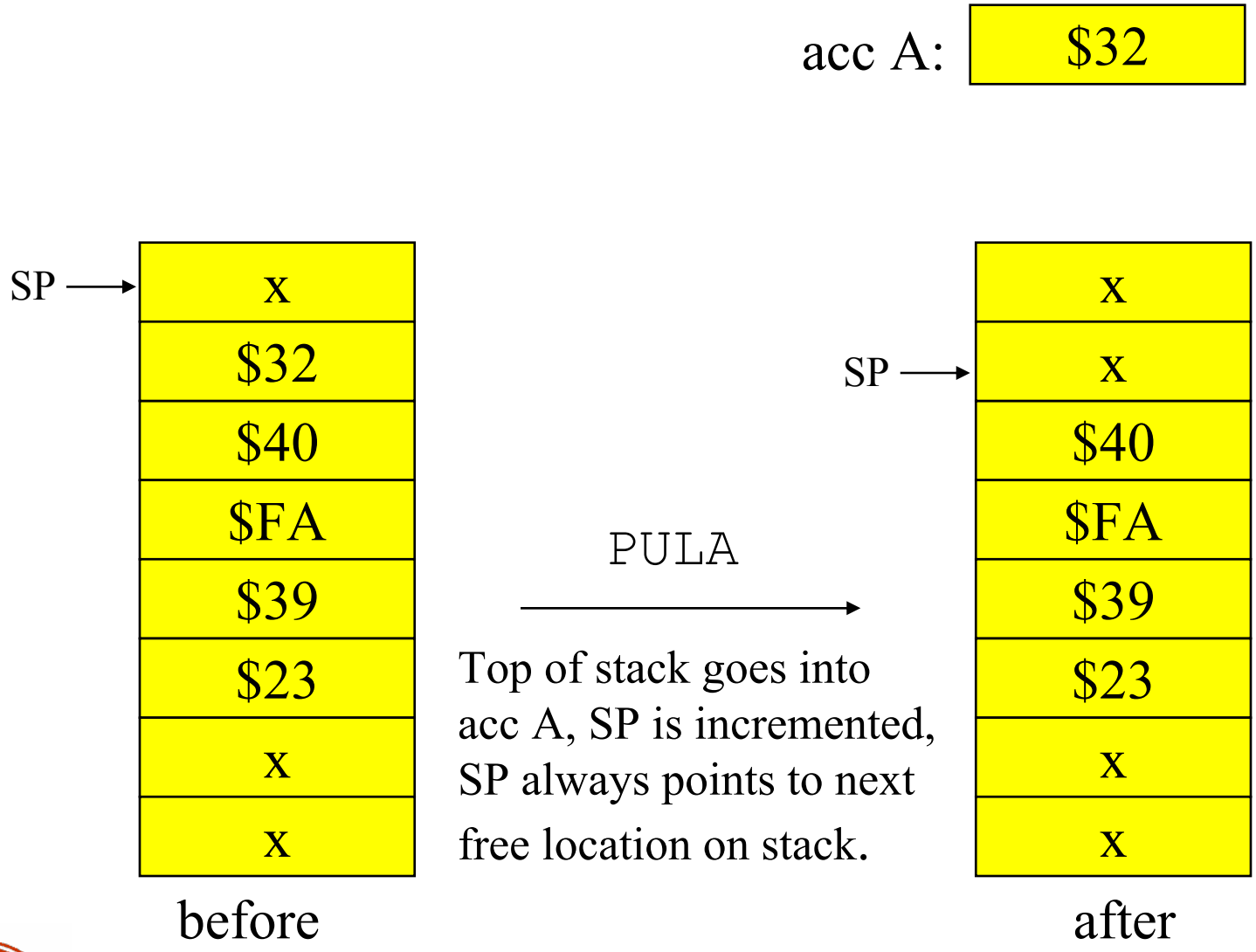
Carlos E. Davila, Electrical Engineering Dept.
Southern Methodist University

The Stack



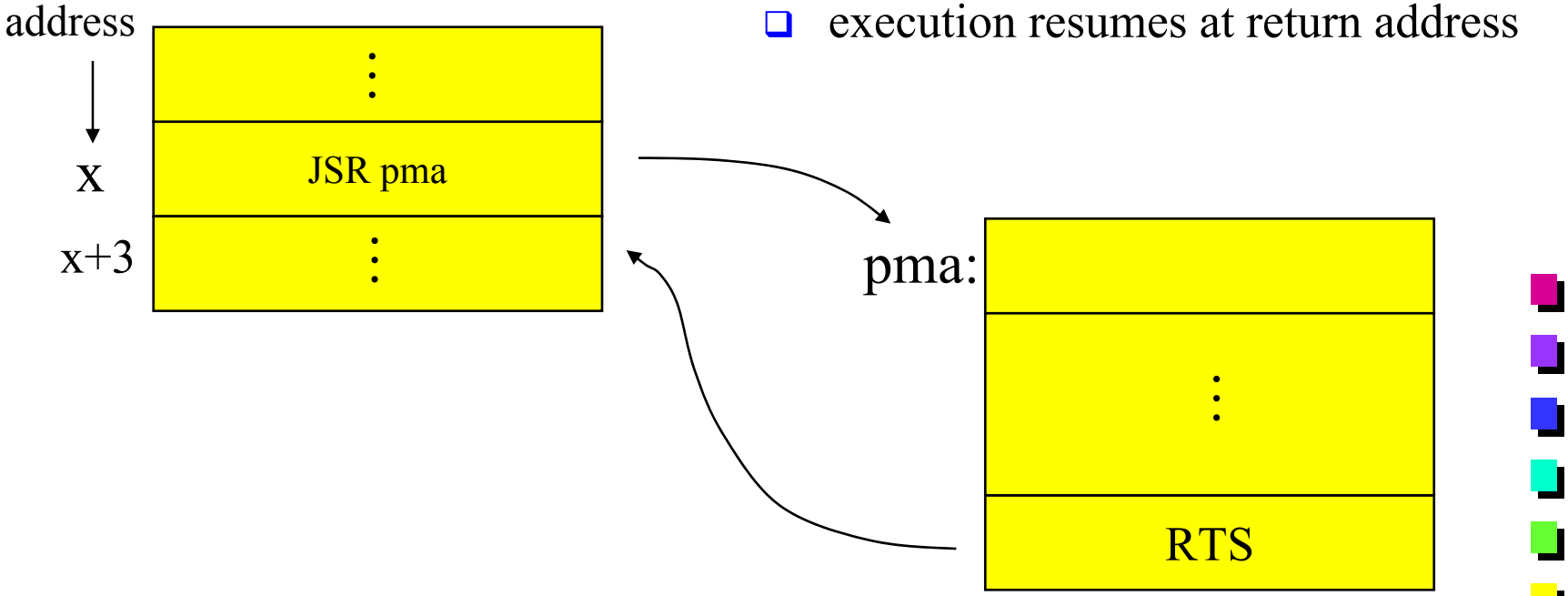
- can "PUSH" a data word on stack from the top, all other words are shifted down one position.
- can "PULL" the top data word from the stack, all other words get shifted up one position.
- can push any number of words.
- stack pointer (SP) always points to top of stack, is initialized by one of the first instructions in program


acc A:
$32
SP
x
x
$40
$FA
$39
$23
x
x
before
PSHA
Pushes contents
of acc A onto stack,
SP is decremented,
SP always points to next
free location on stack.
SP
x
$32
$40
$FA
$39
$23
x
x
after


acc A:
$32
SP
x
$32
$40
$FA
$39
$23
x
x
before
PULA
Top of stack goes into acc A, SP is incremented, SP always points to next free location on stack.
SP
x
x
$40
$FA
$39
$23
x
x
after

Subroutines

- "JSR pma" causes address of the next instruction after the JSR to be pushed on the stack (low byte first).
- PC then loaded with "pma" and execution resumes there.
- "RTS" pulls the return address from stack and loads it into PC
- execution resumes at return address



Interrupts

- Are generated by periferals that must perform time-critical functions.
- A/D converter, serial port, and timer on the 68HC11 can be programmed to generate interrupts.
- Maskable Interrupt: interrupt can be disabled.
- Non-Maskable Interrupt: cannot be disabled.
- 2 bits in the CCR control all interrupts:
  - I: interrupt mask bit, disables (enables) maskable interrupts if = 1 (0)
  - X: if = 1 (0), XIRQ interrupt is disabled (enabled). XIRQ is an interrupt which results when the XIRQ pin is pulled low.
- Individual interrupts can also be masked (disabled) from periferal control registers

When an interrupt occurs:

- Current instruction completes normally.
- Address of next instruction (current value of PC) pushed on stack.
- All CPU registers pushed on stack.
- Maksable interrupts and XIRQ interrupts are disabled. (I, X = 1 in CCR).
- Execution then begins at an address called the interrupt vector (the address of the interrupt service routine, or ISR).
- At the end of the ISR, the RTI instruction causes the saved registers to be pulled from the stack in reverse order, ending with the PC.
- Execution resumes at address of PC prior to interrupt.
- The interrupt vector for a particular periferal is stored in a specific memory location.

Timer interrupts versus polling

- The timer counter (TCNT) is a free-running counter that counts up at a set rate, overflows (goes from FFFF to 0000), then resumes counting.
- The TCNT can be programmed to generate an interrupt each time it overflows. The ISR can then be used to perform an analog to digital conversion at regular sampling intervals.
- Alternately, the TCNT can be “polled”. This is done by disabling timer interrupts and monitoring the TOF bit in the timer interrupt flag register 2 (TFLG2). TOF is set (=1) each time the TCNT overflows. By monitoring TOF, A/D conversions can be made at regular intervals. The TOF bit must be cleared by software. Use TST instruction and check if result is negative.

A/D Converter on the 68HC11

- A/D control/status register (ADCTL) : each time ADCTL is written to, an A/D conversion begins.
- CCF bit in ADCTL can be used to test when conversion is complete.
- SCAN bit in ADCTL should be cleared (=0).
- MULT bit should be cleared.
- CD, CC, CB, and CA should be cleared so that channel PE0 is sampled. Result appears in result register ADR1.
- See users manual for address of all pertinent registers.