

ECE/CS 5780-6780 : Embedded Systems Design

Lect. 05: *Memory manipulation - Interrupts*

Pierre-Emmanuel Gaillardon

Department of Electrical and Computer Engineering – University of Utah



Spring 2017
Salt Lake City, UT, USA





Objectives for Today!

- **Get familiar with the ARM M processor architecture**
- **Understand the architecture of a microcontroller**
- Acquire the fundamentals of the hardware/software interface
- Acquire the fundamentals for sensing and controlling the physical world
- Learn modeling techniques for embedded system design
- **Understand the usage of several peripherals** through labs
- Design a complete embedded system – from specs to realization
 - Realization of a PCB
 - Behavioral modeling of the system
 - Complete SW realization



Memory Stack



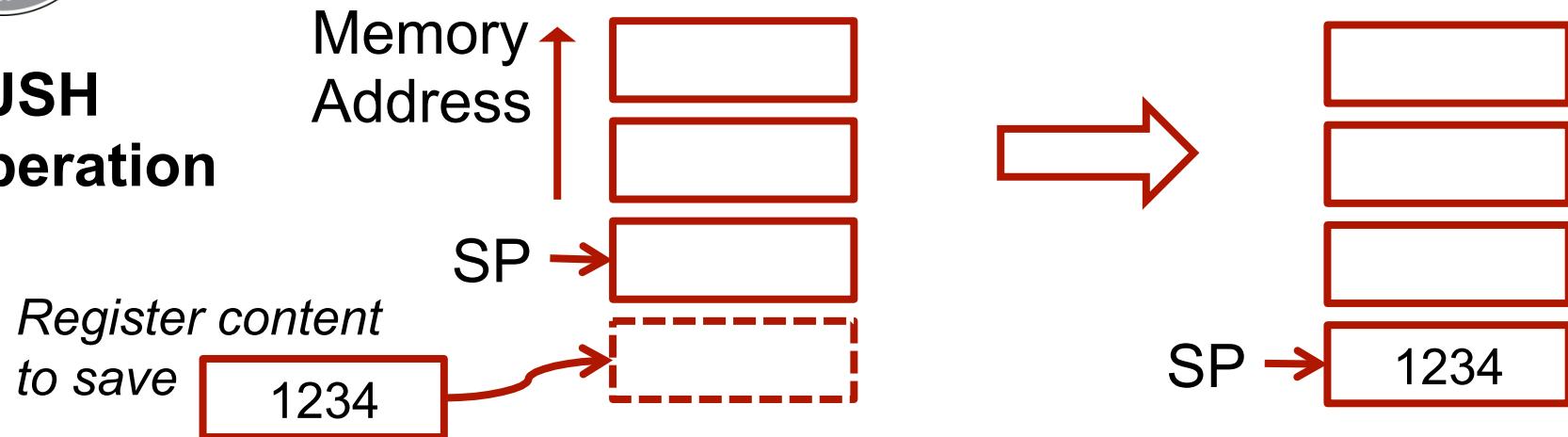
Stack Memory Operations

- Stack memory is a memory usage that allows the system memory to be used as temporary data storage.
- Particularly useful for register storage.
- Behaves as a first-in last-out buffer.
- Cortex-M uses a “full-descending” stack model.
- Storing register to the stack is called PUSH.
- Restoring register from the stack is called POP.
- SP (R13) register indicates where the current stack memory location is.



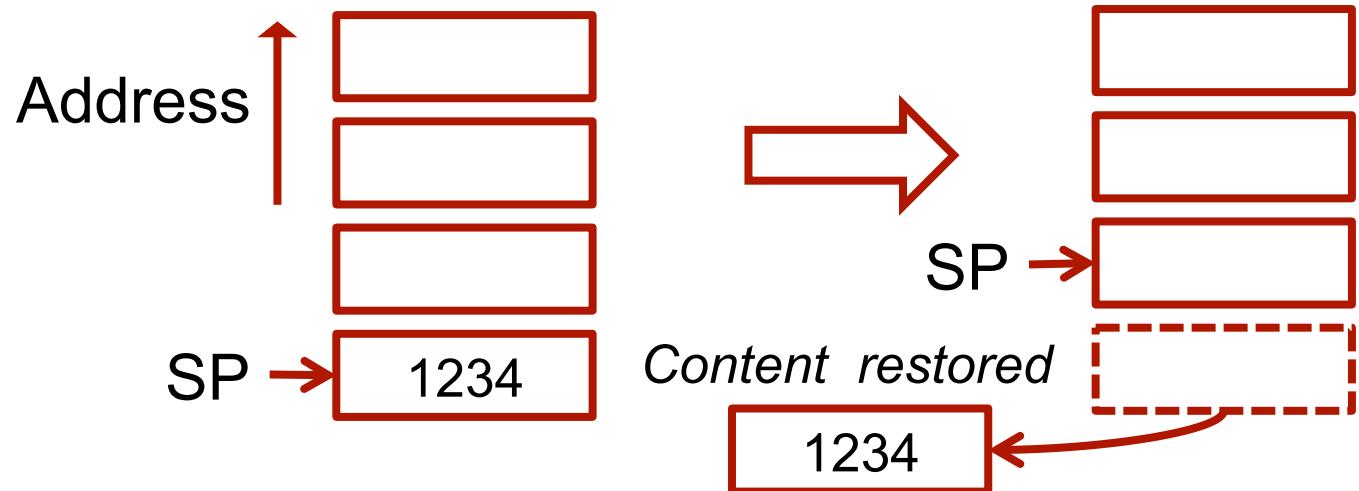
PUSH and POP

PUSH Operation



Data processing (Original content gets destroyed)

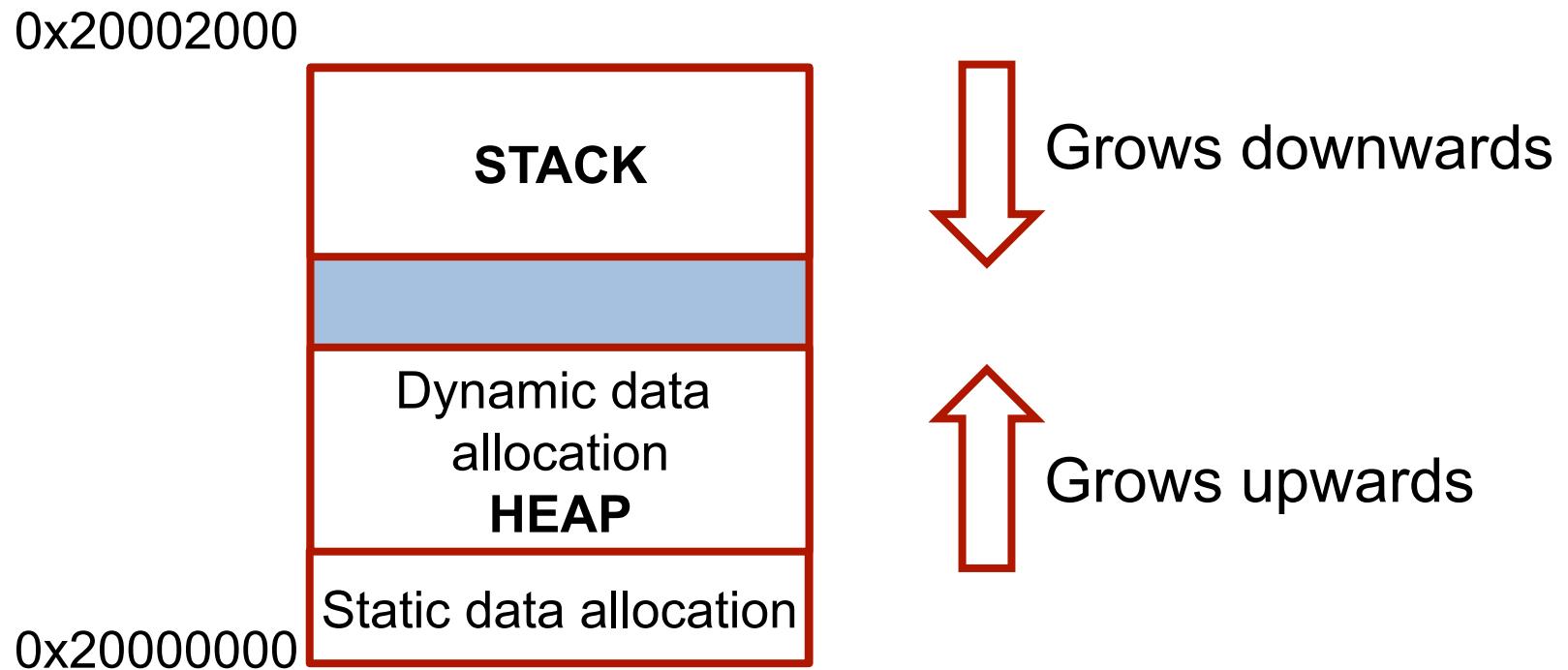
POP Operation





Why does the Stack grow downwards?

To allow maximum flexibility!



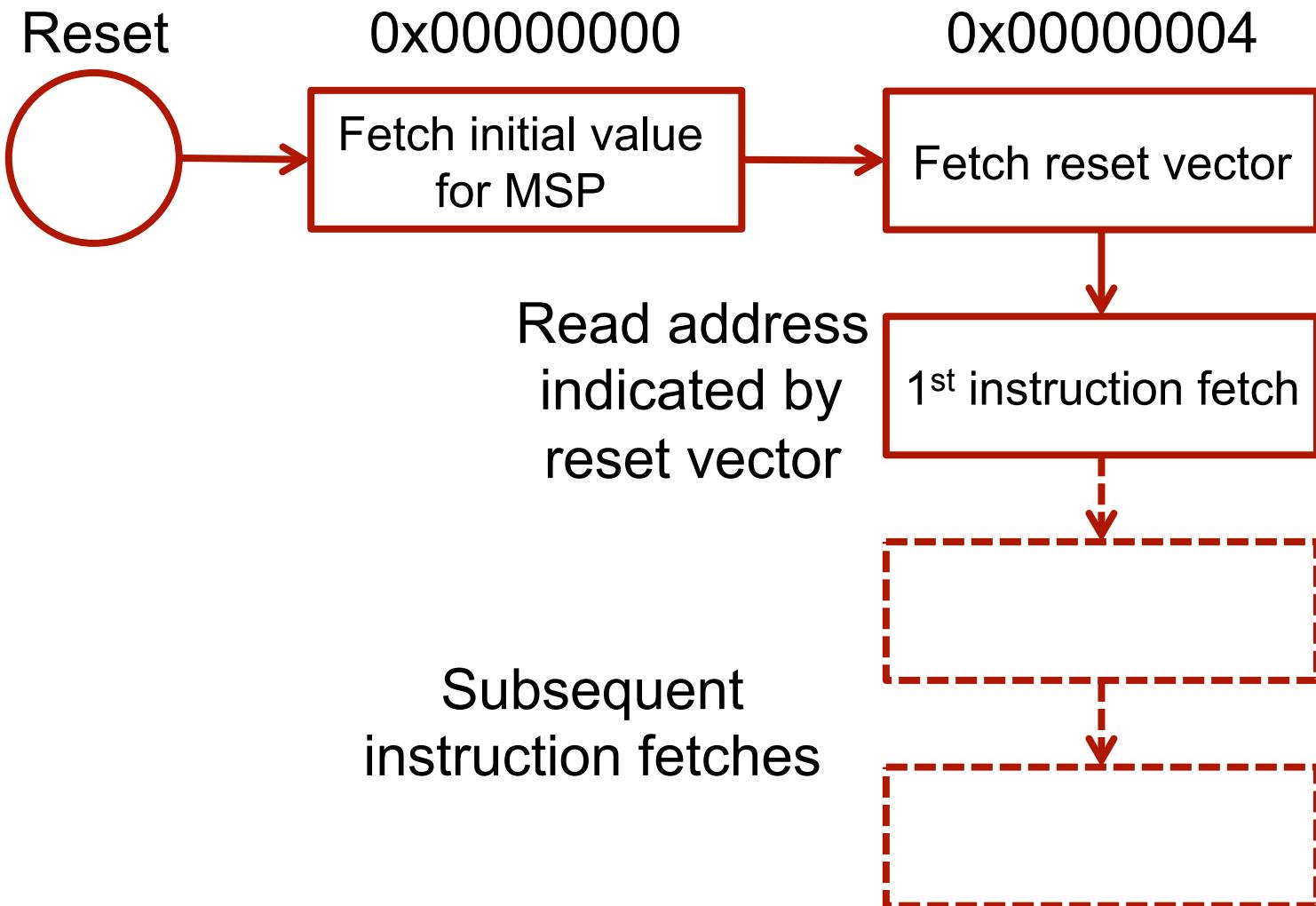
On a general basis, it is dangerous to use dynamic data allocation!



Processor-startup

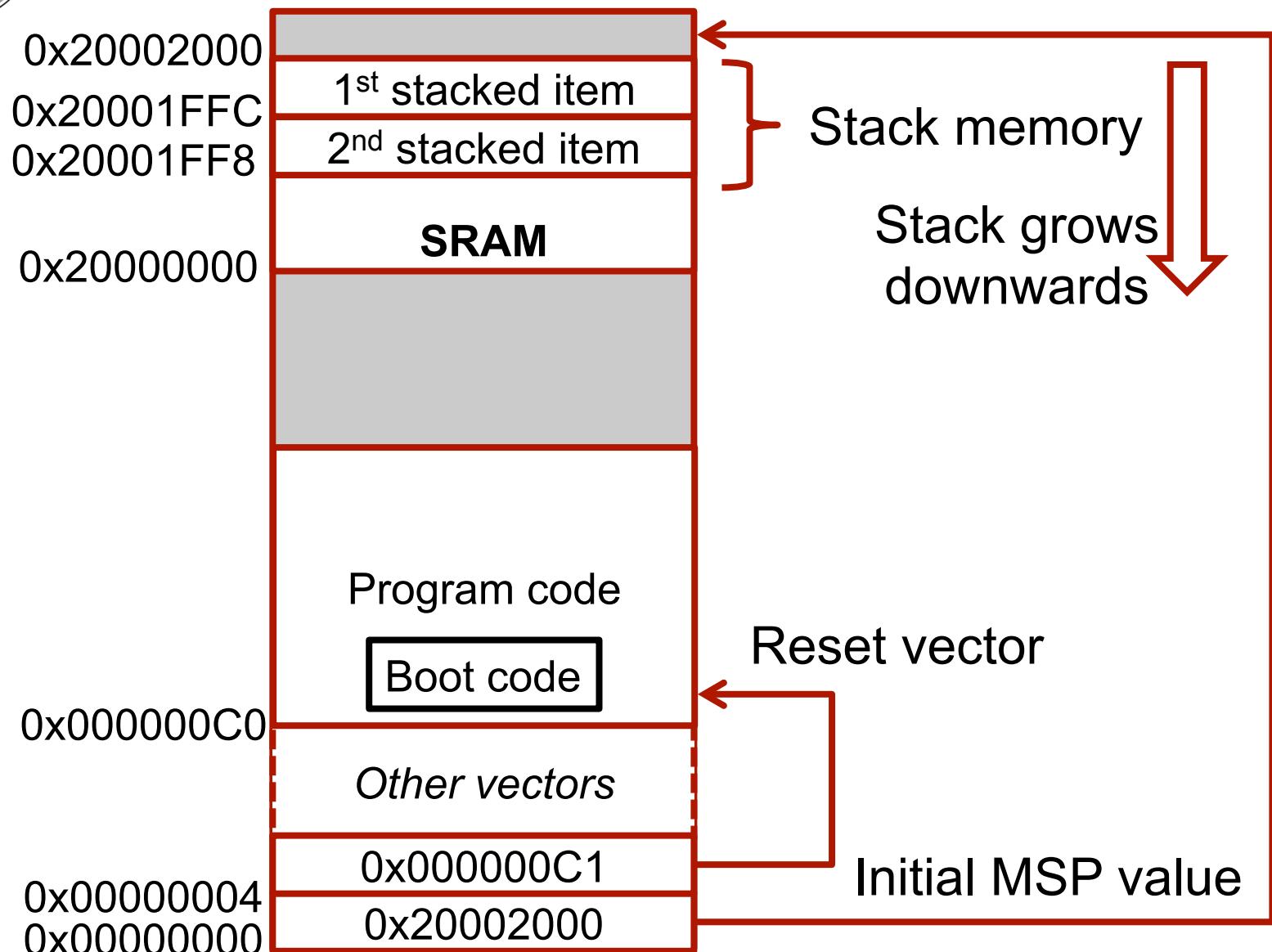


Reset Sequence





Example of MSP and PC Initialization





Memory-manipulation in Embedded Software



Register Access

- Memory has an address and value
- Can equate a pointer to desired address
- Can set/get de-referenced value to change memory

```
#define SYSREG_SOFT_RST_CR 0xE0042030  
  
uint32_t *reg = (uint32_t *)(SYSREG_SOFT_RST_CR);  
  
main () {  
    *reg |= 0x00004000; //ResetGPIOhardware  
    *reg &= ~(0x00004000);  
}
```



What Happens when you R/W a pointer?

```
#include <stdio.h>
#include <inttypes.h>

#define REG_FOO 0x40000140

Main () {
    uint32_t *reg = (uint32_t *)(REG_FOO);
    *reg+=3;
    printf("0x%x\n",8*reg);// Prints out new value
}
```

Thank you for your attention

Questions?



Laboratory for NanoIntegrated Systems

Department of Electrical and Computer Engineering

MEB building – University of Utah – Salt Lake City – UT – USA