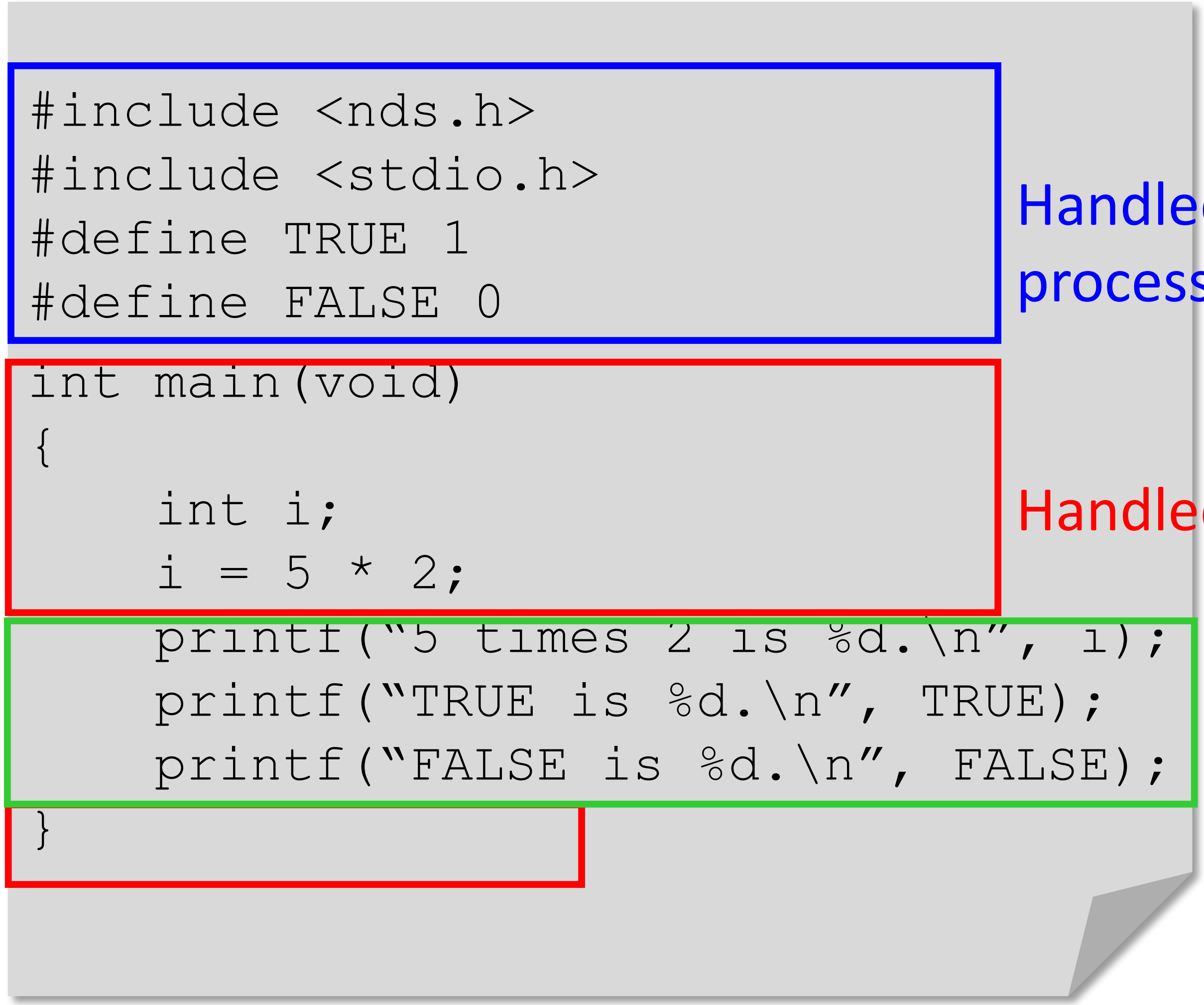
Review

- Datasheets

Example: compiling C source code



```
#include <nds.h>
#include <stdio.h>
#define TRUE 1
#define FALSE 0
int main(void)
{
    int i;
    i = 5 * 2;
    printf(“5 times 2 is %d.\n”, i);
    printf(“TRUE is %d.\n”, TRUE);
    printf(“FALSE is %d.\n”, FALSE);
}
```

Handled by the pre-processor

Handled by the compiler

Implemented in C library for the target platform

Memory Stack

- Tempory data storage
- First in the bottom of the stack is the last one out

Stack Memory Operations

- Stack memory is a memory usage that allows the system memory to be used as temporary data storage.

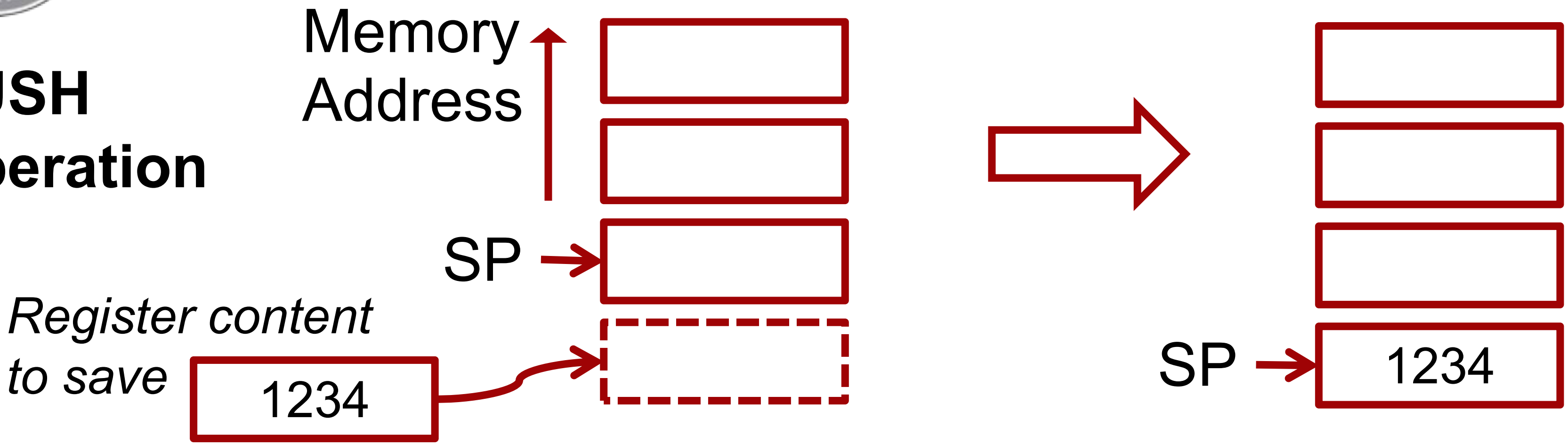
- Particularly useful for register storage.
- Behaves as a first-in last-out buffer. FILO
- Cortex-M uses a “full-descending” stack model.

- 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



Memory Address

SP

Register content to save

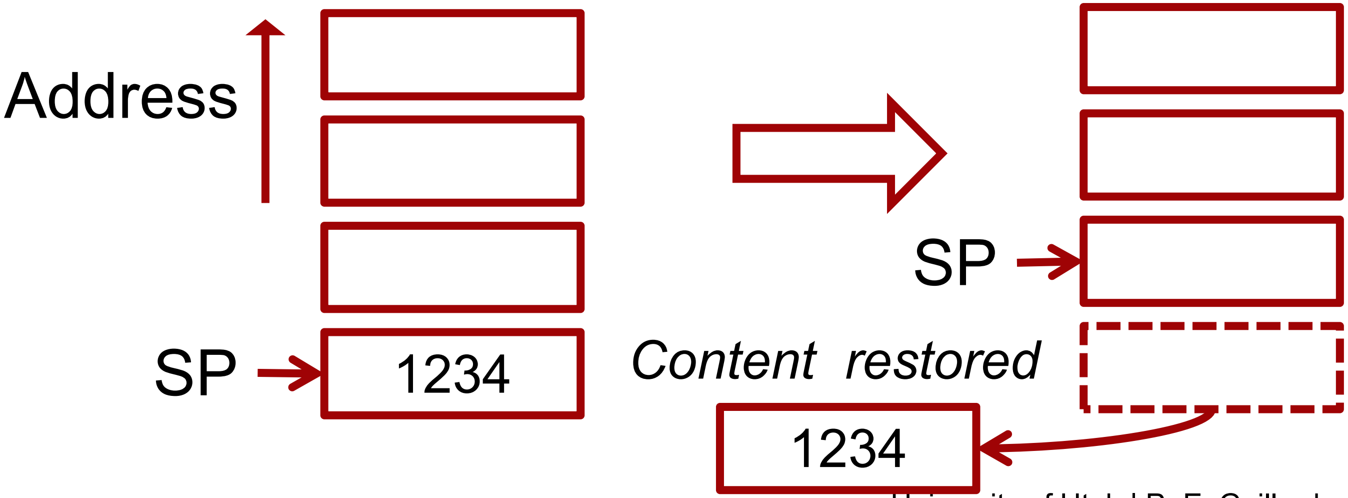
1234

SP

1234

Data processing (Original content gets destroyed)

POP Operation



Address

SP

1234

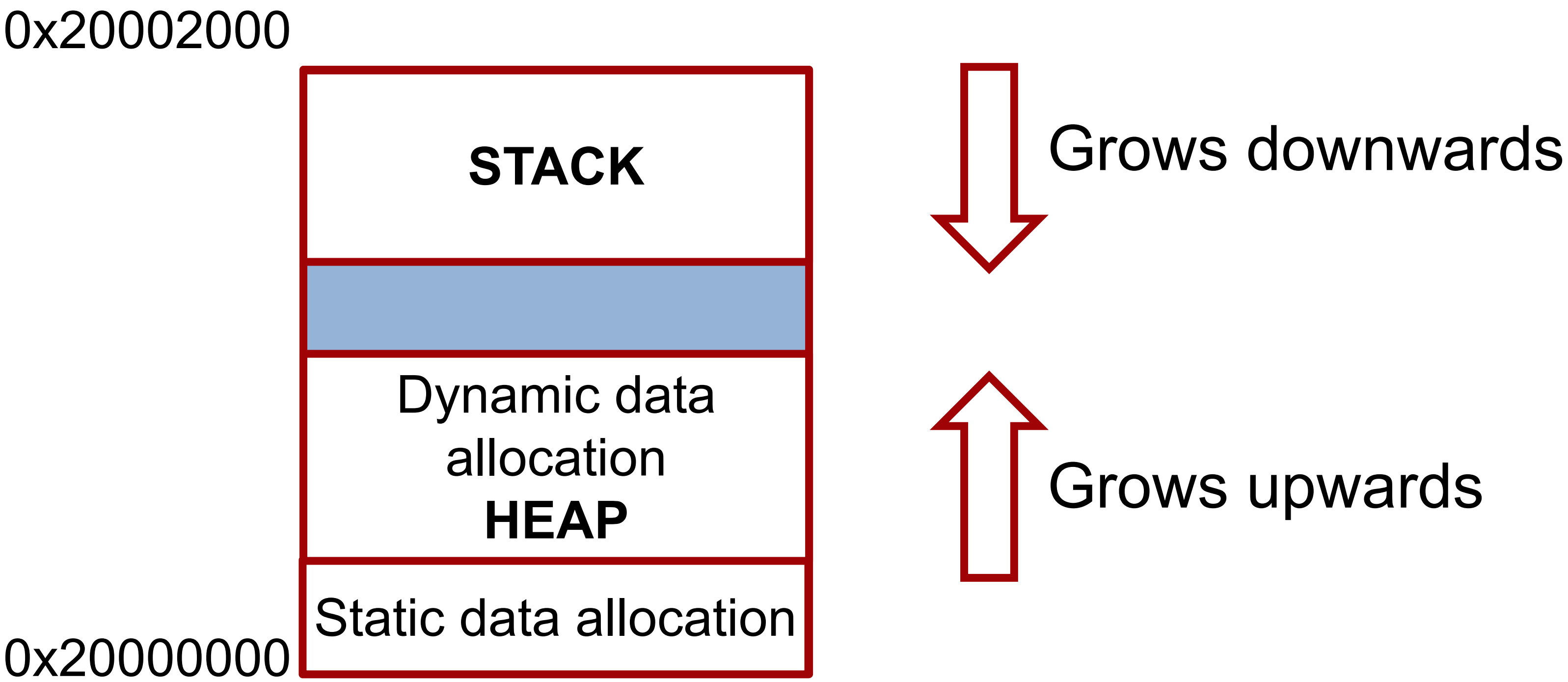
SP

Content restored

1234

Why does the Stack grow downwards?

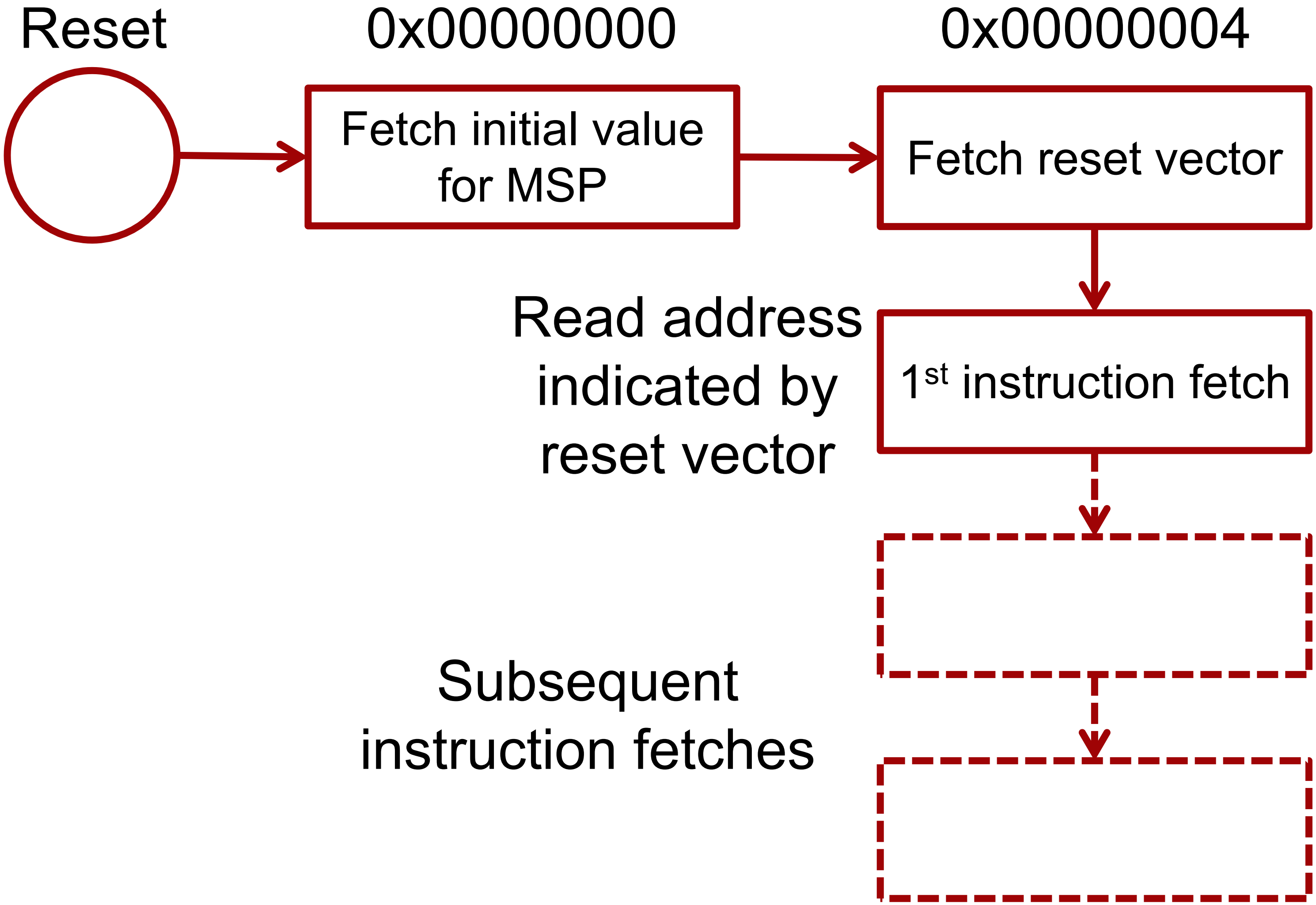
To allow maximum flexibility!



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

Processor-startup

Reset Sequence



Reset

0x00000000

Fetch initial value for MSP

0x00000004

Fetch reset vector

Read address indicated by reset vector

1st instruction fetch

Subsequent instruction fetches