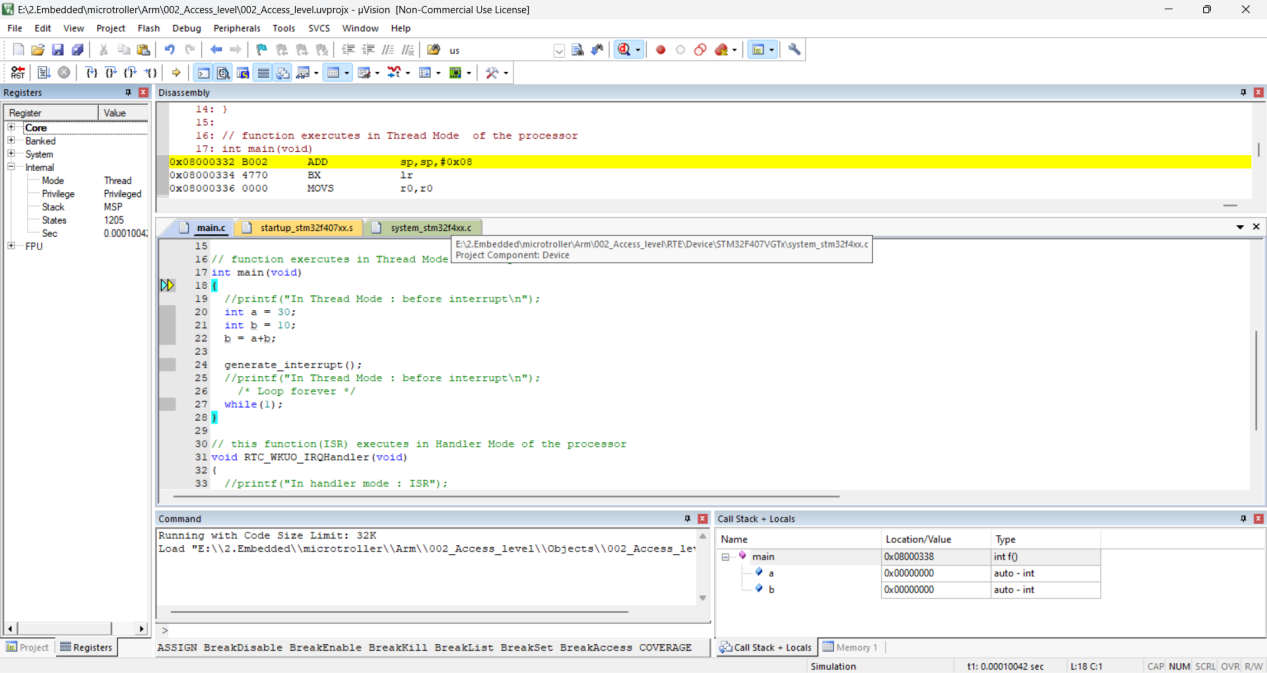
Arm Cortex M3/M4

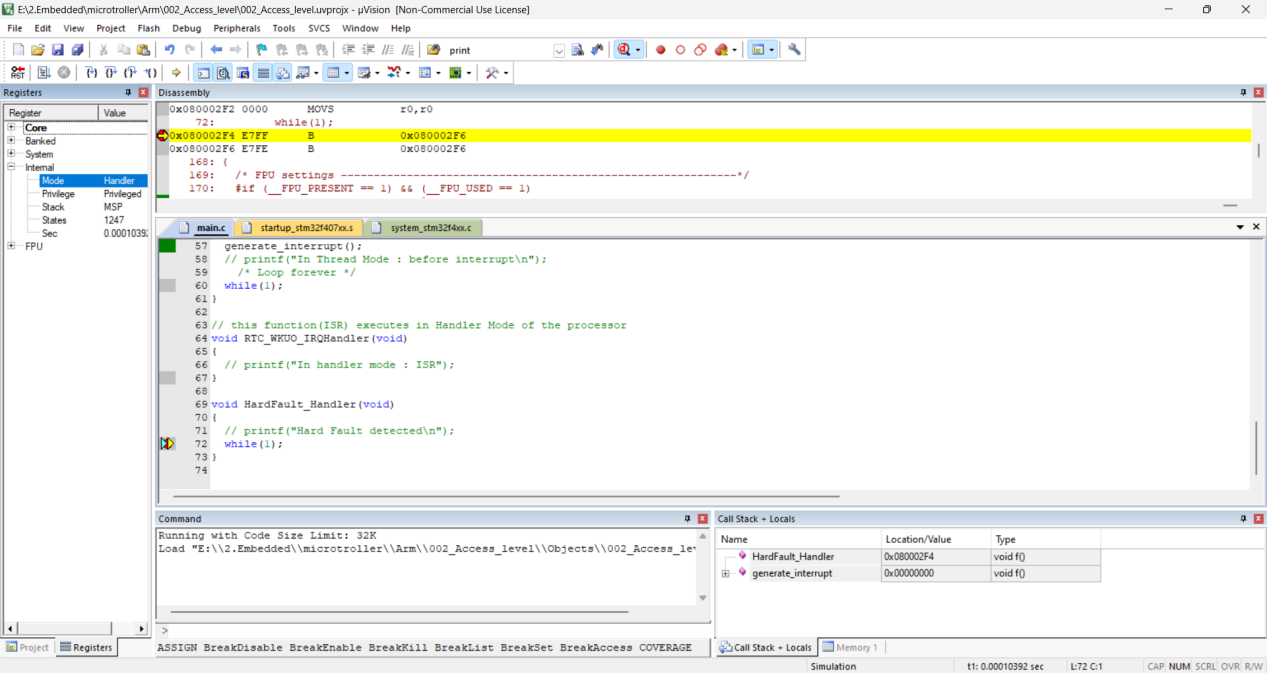
1. Operational mode of the processor

Đối với ARM core Mx có 2 lại chế độ hoạt động: Thread mode(User Mode) và Handler mode

- vi sử lí luôn bắt đầu ở chế độ Thread mode:



- chế độ handler mode chỉ được kích hoạt khi xuất hiện Exception/interrupt event:



1. Access level of the processor

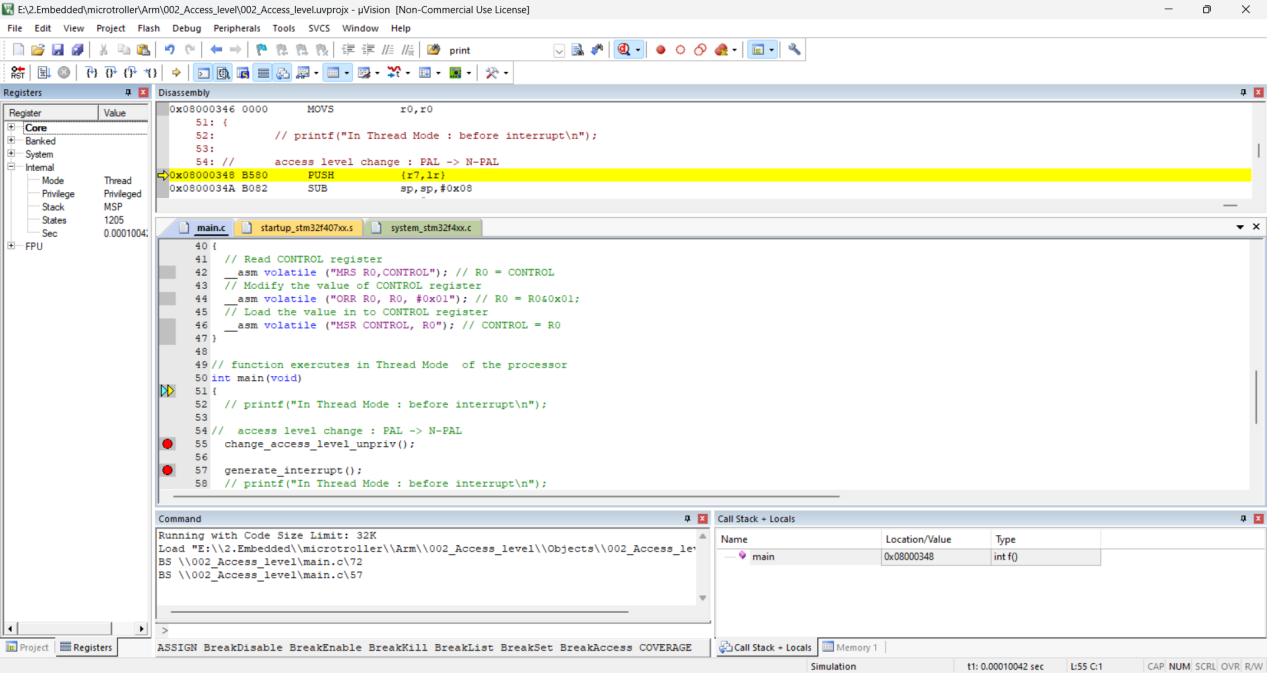
- ARM cortex có 2 cấp truy cập PAL(Privileged Access Levels) và NPAL(Non-Privileged Access Levels)

- Nếu code thực thi ở cấp truy cập PAL, vi sử lí có thể truy cập toàn bộ nguồn dự liệu và các thanh ghi

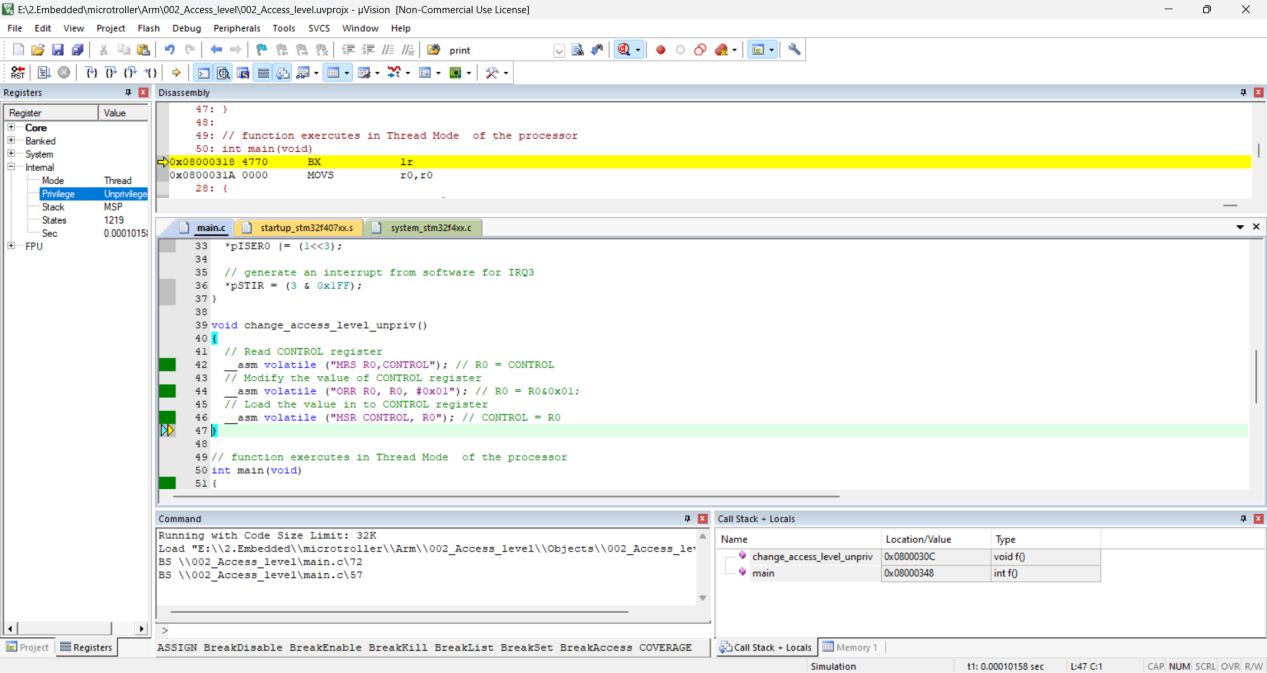
- Nếu code thực thi ở cấp truy cập NPAL, vi sử lí không thể truy cập một vài thanh ghi hạn chế

- PAL là cấp truy cập mắc định của vi sử lí

- Thread Mode có thể hoạt động ở 2 cấp truy cập PAL và NPAL.



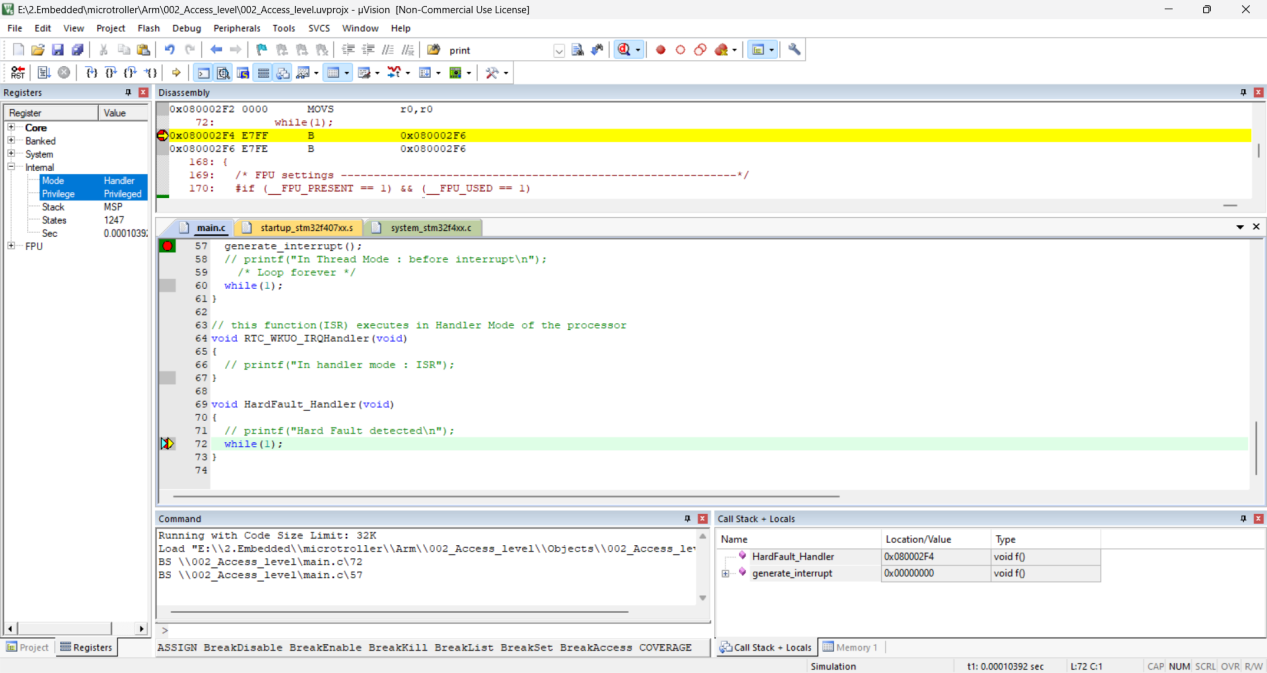
- chuyển từ PAL -> NPAL bằng cách set bit nPRIV trong CONTROL register



- Khi set bit nPRIV không thể clear nPRIV(tức là ở chế độ thread mode khi đã lựa chọn NPAL không thể sửa lại được).khi cố tình sửa lại bit nPRIV hoặc truy cập vào các thanh ghi bị giới hạn sẽ sảy ra lỗi hard fault

- cách duy nhất có thể chuyển từ NPAL-> PAL là chuyển từ Thread mode sang Handler mode

-Handler Moden luôn luôn thực thi code tại PAL



3 thanh ghi của Cortex Mx(32 bit)

- Các thanh ghi dùng chung bao gồm R0-R12

- Thanh ghi R13 là SP(Stack pointer) giữ địa chỉ của đầu Stack

- LR(link register) nhận địa chỉ trả về từ PC(Program Counter). Khi thoát khỏi hàm hoặc là các ngắt (interrupts/exception) thì PC -> LR

- PC(Program Counter) thanh ghi chứa địa chỉ của câu lệnh kế tiếp được thực thi

- Program Status Register

Program Status Register

T bit của EPSR

- Nhiều bộ xử lý ARM hỗ trợ sự tương tác ARM - Thumb. Tức là có thể chuyển đổi giữa 2 trạng thái ARM và Thumb

- Bộ xử lý phải ở trạng Thái ARM để thực thi các lệnh từ ARM ISA

- Bộ xử lý phải ở trạng Thái Thumb để thực thi các lệnh từ Thumb ISA

- Nếu T bit (EPSR) được set(1), bộ vi xử lý sẽ thực thi lệnh từ Thumb ISA

- Nếu T bit (EPSR) được reset(0), bộ vi xử lý sẽ thực thi lệnh từ ARM ISA

- Đối với bộ xử lý ARM Cortex Mx không hỗ trợ trạng thái ARM -> T bit luôn được đặt ở giá trị 1 . Khi T bit = 0 -> sảy ra Usage fault exception

- lsb(bit 0) của PC được liên kết với T bit. Khi load địa chỉ vào PC thì bit cuối cùng được nạp vào T bit -> 0th bit của địa chỉ luôn luôn là 1

-> đó là lý do tại sao toàn bộ địa chỉ vector được tăng lên 1 trong vector table

4 ARM gcc inline assembly code

- Inline assembly code được sử dụng để viết code assembly trong ngôn ngữ C

5 các bước Reset của bộ xử lý ARM

- Khi vi xử lý reset, PC load giá trị 0x0000\_0000

- Processor đọc giá trị địa chỉ tại địa chỉ 0x0000\_0000 vào MSP(main stack pointer register) PC nhảy đến địa chỉ tếp theo(0x0000\_0004) -> processor khởi tạo giá trị của stack pointer

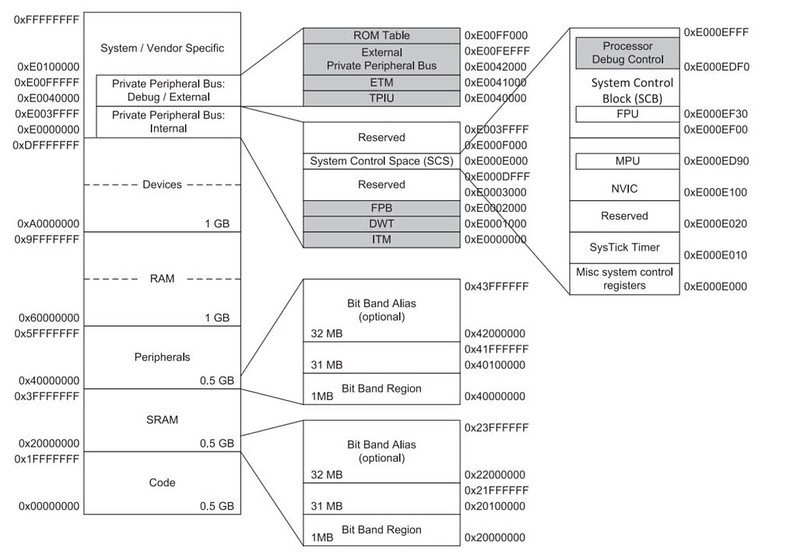
- sau khi Processor đọc giá trị tại 0x0000\_0004 vào PC giá trị đại chỉ của reset Handler

- PC nhảy đến reset handler

- trong Reset handler được viết bởi hàm C hoặc Assembly để tiến hành và khởi tạo các thứ cần thiết

- cuối cùng set gọi hàm main()

6 Memory Map of the processor



- memory map giải thích sự ánh xạ của các thanh ghi ngoại vi và bộ nhớ

- Bộ xử lý, phạm vi vị trí bộ nhớ có thể định địa chỉ, phụ thuộc vào kích thước của bus địa chỉ

\* Code region:

- Đây là khu vực mà các nhà cung cấp MCU nên kết nối bộ nhớ CODE

- Các loại bộ nhớ Code : Flash, ROM, OTP ,EEPROM, etc

- Bộ xử lý theo mặc định sẽ lấy vector ngay sau khi reset

\*SRAM region

7 Stack memory

- Stack memory là một phần của bộ nhớ chính ( RAM nội và ngoại) được dành riêng cho việc lưu chữ dữ liệu tạm thời

- sử dụng chín trong hàm, xử lý ngắt và ngoại lệ

- stack memory được truy cập theo kểu LIFO(last in first out). Có thể truy cập khi sử dụng PUSH và POP và được sử dụng bất kỳ hướng dẫn thao tác bộ nhớ nào LD, STR

- Stack được theo dõi bằng thanh ghi SP. lệnh PUSH và POP ảnh hưởng đến thanh ghi SP (tăng hoặc giảm)

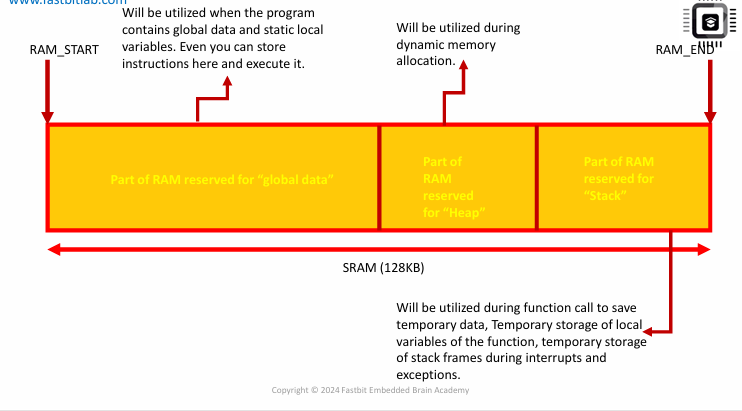
Sử dụng Stack memory

- lưu trữ tạm thời các giá trị thanh ghi bộ xử lý

- lưu trữ gias trị cụ bộ của hàm

- trong việc xử lý ngoại lệ và ngắt, Stack memory sẽ được dùng để giữ nội dung(các giá thị thanh ghi dùng trung, trạng thái của thanh ghi, giá trị trả về) của các mã đang thực thi

Kiến trúc bộ nhớ RAM



Global data

Heap

Stack

Mô hình hoạt động ngăn xếp

- Full Ascending stack (FA)

- Full Descending stack(FD) (ARM Cortex Mx processors use this)

- Empty Ascending stack(EA)

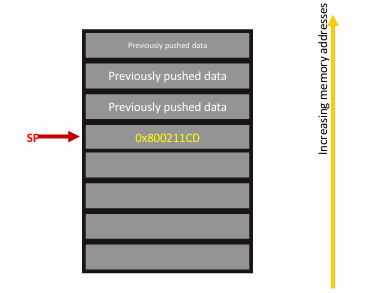
- Empty Descending stack(ED)

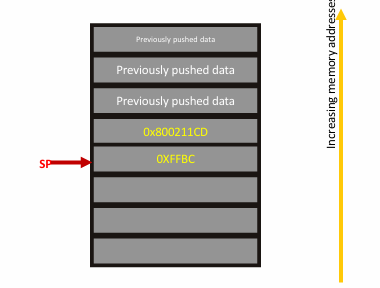
Full Descending stack(FD)

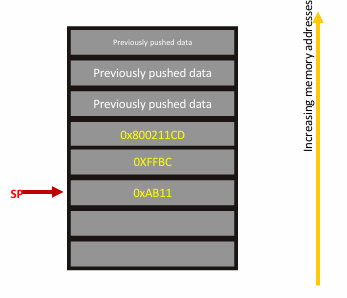
|  |  |
| --- | --- |
|  | Các đặc điểm của FD  -SP bắt đầu giữ một giá trị địa chỉ  - Khi thực hiện lệnh PUSH thì giá trị địa chỉ của SP giảm , địa chỉ SP sẽ giữ giá trị đã được PUSH lên  - khi thực hiện lênh POP thì giá trị của SP sẽ tăng |

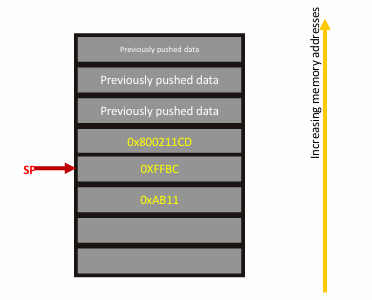
EX : thực hiện các lệnh

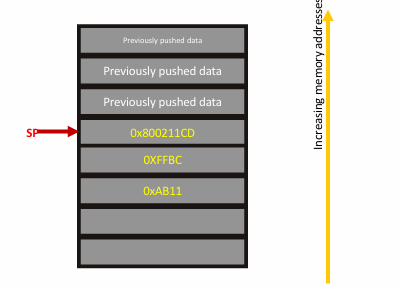
|  |  |
| --- | --- |
|  |  |

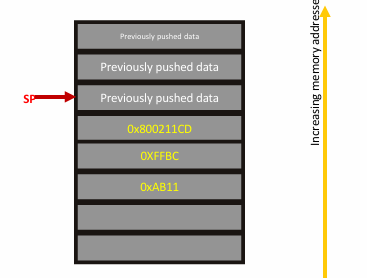












Stack placement

