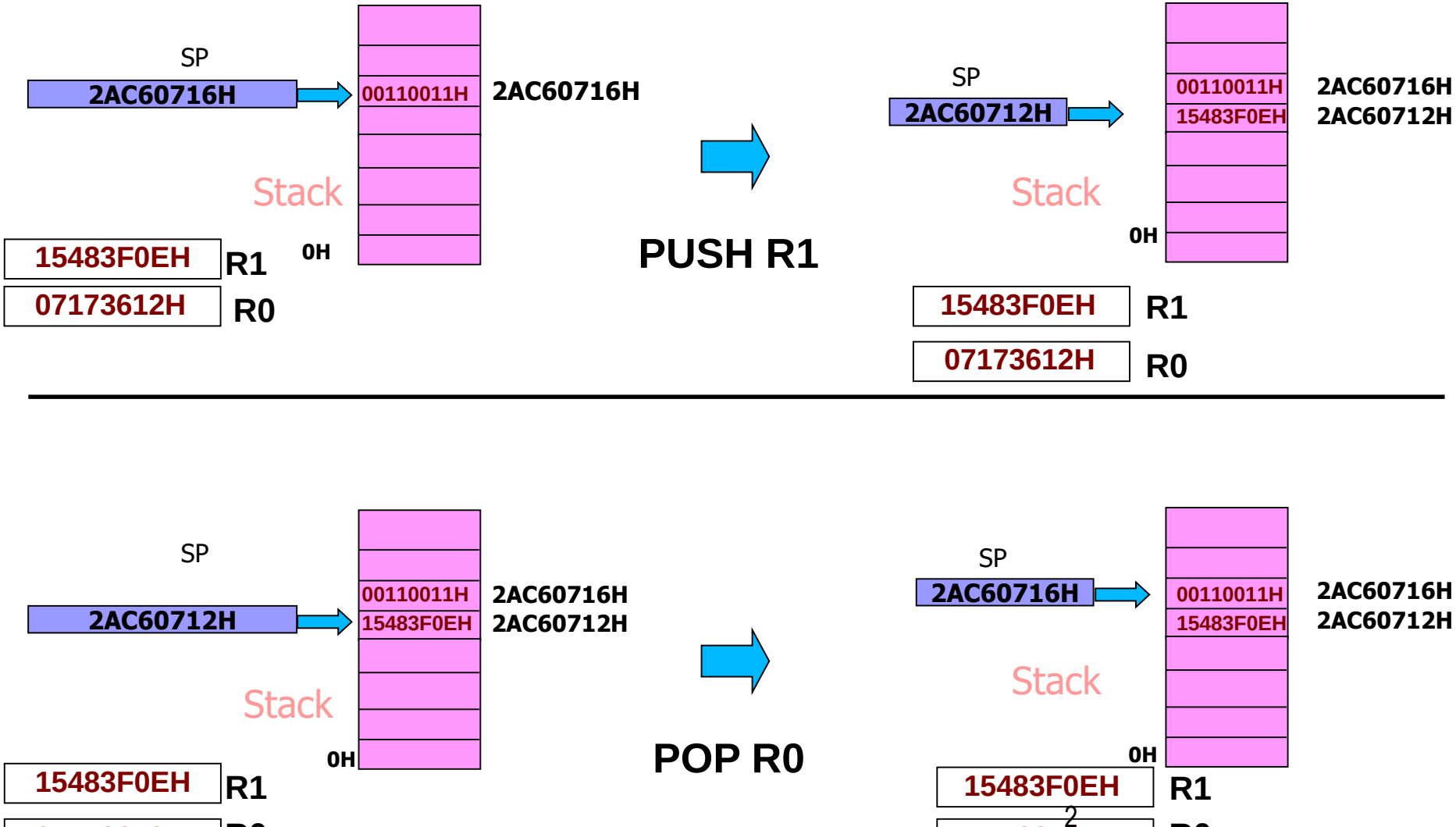
Tema 2

Uso de la Pila

2 Pila (Stack)

- Operaciones basicas de la pila



2 Pila (Stack)

Operaciones basicas de la Pila

- Los datos almacenados en registros se guardan en la memoria de la pila mediante una operacion PUSH…
- …y puede ser restablecido a registros mas tarde mediante una operacion POP
- El valor del registro SP se ajusta automaticamente en las operaciones PUSH y POP, por lo que multiples PUSH consecutivos no borrarán los datos previamente insertados.
- Importante el orden de PUSH y POP: El orden de **POP debe ser el inverso al orden de PUSH**
- Operaciones donde interviene la Pila, como **Cambio de contexto y Paso de Parámetros**
  - Se simplifican, gracias a las instrucciones PUSH y POP con multiple carga y almacenamiento
  - The procesador automaticamente invierte en POP el orden de la lista de registros de PUSH

2 Pila (Stack)

Implementacion del Stack

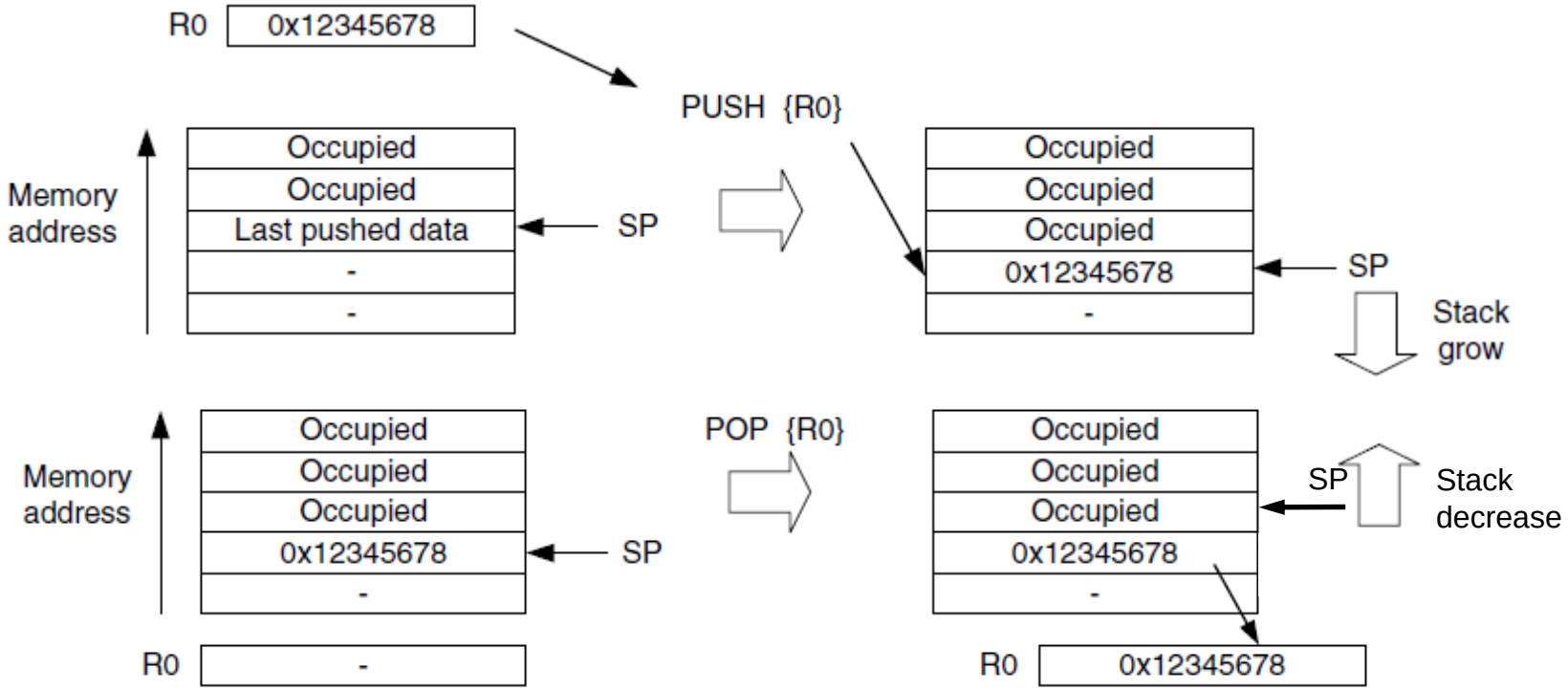
- Las operaciones PUSH y POP solo se pueden hacer sobre registros
- Cada operacion PUSH/POP **transfiere 4 bytes of datos** (la word completa del registro)
- Ademas, el valor del registro SP se decrementa/incrementa en 4 en cada operacion PUSH/POP (respectivamente) o en un multiplo de 4 si se apilan mas de 1 registro

Uso de la pila en Manejo de Excepciones (ISR)

- Cuando se entra a una ISR, **un numero de registros se almacenan en la pila automaticamente**, y R13 se usará como registro SP para este proceso
- De foma analoga, **los registros almacenados se restableceran automáticamente cuando se salga de la ISR**, y el registro SP tambien será adjustado.

2 Pila (Stack)

- El Cortex-M3 utiliza un modelo de operacion con la pila full-descending
  - **SP apunta al ultimo dato introducido en el** stack
  - SP en la operacion PUSH se **decrementa antes de almacenar en la pila, y se incrementa en la operacion POP despues de sacar el dato de la pila**



2 Pila (Stack)

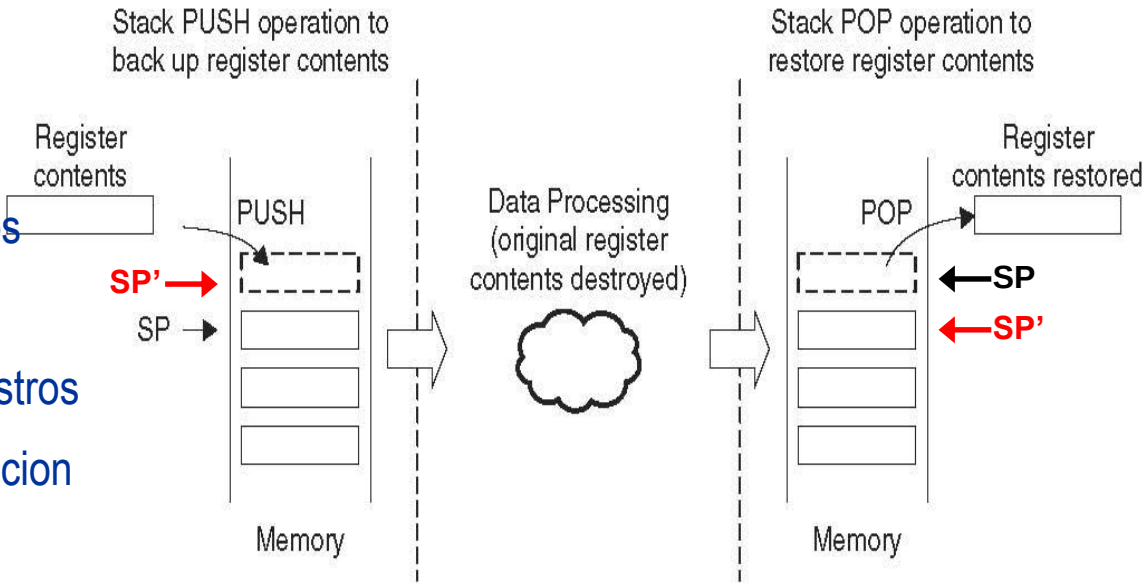
- **Guardando en la pila un solo registro:**

  PUSH {R0} ; R13 = R13-4, y luego Mem[R13] = R0

  POP {R0} ; R0 = Mem[R13], y luego R13 = R13+4

- **Llamada a subrutina:**

```
subroutine_1
   PUSH {R0-R7, R14} ; Guarda registros
   …                 ; Procesado
   POP {R0-R7, R14}  ; Restablece registros
   BX R14            ; Retorna a la funcion
```



2 Pila (Stack)

Ejemplo de uso de pila

- **Llamada a subrutina**

Main program

```
    ...
; R0   X, R1   Y, R2   Z
BL    function1
```

Subroutine

```
function1
    PUSH    {R0} ; store R0 to stack & adjust SP
    PUSH    {R1} ; store R1 to stack & adjust SP
    PUSH    {R2} ; store R2 to stack & adjust SP
    ... ; Executing task (R0, R1 and R2
      ; could be changed)
    POP     {R2} ; restore R2 and SP re adjusted
    POP     {R1} ; restore R1 and SP re adjusted
    POP     {R0} ; restore R0 and SP re adjusted
    BX      LR   ; Return
```

```
; Back to main program
; R0   X, R1   Y, R2   Z
... ; next instructions
```

2 Pila (Stack)

Ejemplos de uso de pila

- Llamada a rutina con instrucciones PUSH/POP con multiple carga y restablecimiento

Main program

```
    ...
; R0   X, R1   Y, R2   Z
BL    function 1
```

Subroutine

```
function 1
    PUSH    {R0 R2} ; Store R0, R1, R2 to stack
    ... ; Executing task (R0, R1 and R2
      ; could be changed)
    POP     {R0 R2} ; restore R0, R1, R2
    BX      LR   ; Return
```

```
; Back to main program
; R0   X, R1   Y, R2   Z
... ; next instructions
```

- Para subrutinas (functiones) anidadas LR tambien debe ser metido en la pila

Main program

```
    ...
; R0   X, R1   Y, R2   Z
BL    function 1
```

Subroutine

```
function 1
    PUSH    {R0 R2, LR} ; Save registers
                ; including link register
    ... ; Executing task (R0, R1 and R2
        ; could be changed)
    POP     {R0 R2, PC} ; Restore registers and
                ; return
```

```
; Back to main program
; R0   X, R1   Y, R2   Z
... ; next instructions
```