

Push and Pop

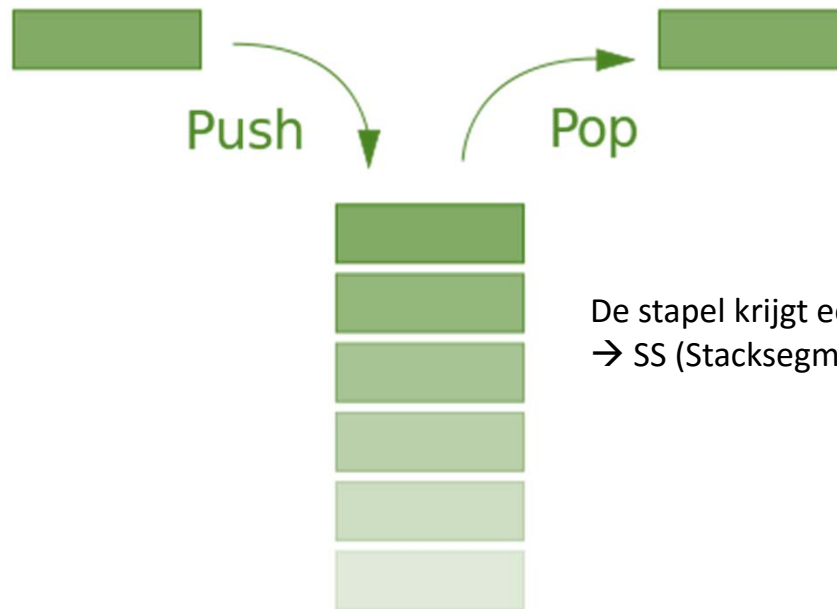
Wat als...

Een register tijdens een bewerking meerder waarden moet bevatten?

→ Oplossing → Gebruik van push/pop

PUSH = De oorspronkelijke waarde wordt op een stack geplaatst. (register is vrij)

POP = Oorspronkelijke waarde terug → waarde terug van de stapel



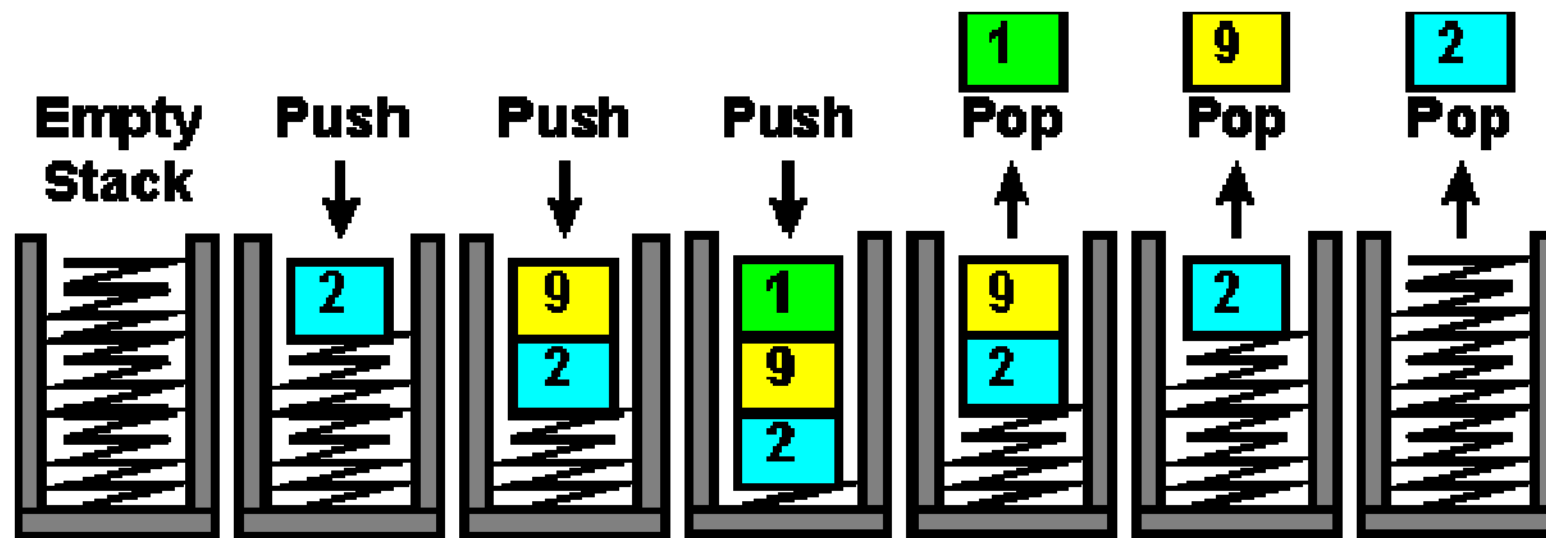
De stapel krijgt een vaste plaats in het geheugen.

→ SS (Stacksegment), op het adres FEEE (aangegeven door de stackpointer)

LIFO (Last in, first out)

Push and Pop

LIFO (Last in, first out)



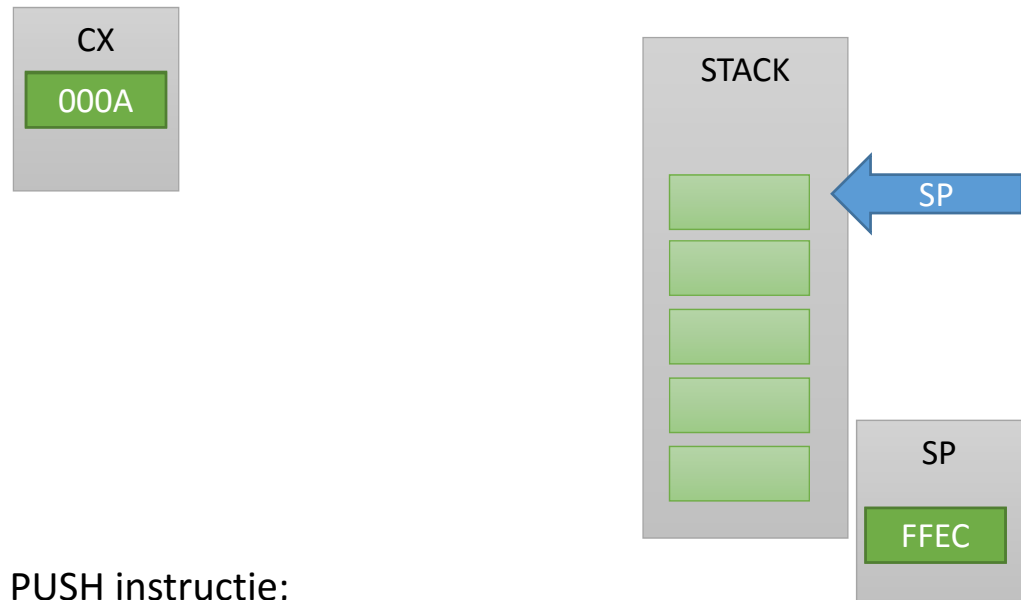
Push and Pop

Voorbeeld PUSH

Het CX register is gevuld met de waarde 000A.

Push instructie: Registerwaard wordt op stack geplaatst.

→ PUSH CX



Na de PUSH instructie;

- Het register is terug vrij voor gebruik.
- De Stackpointer geeft de volgende stacklocatie aan.

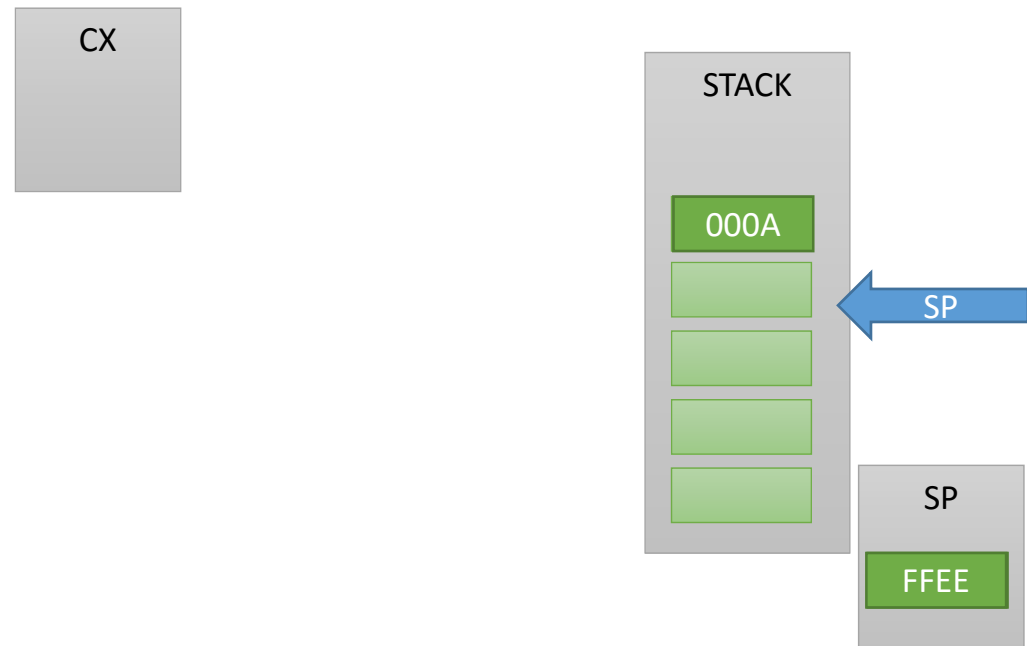
Push and Pop

Voorbeeld POP instructie

Het CX register is gevuld met de waarde 0000

POP: De waarde op stack wordt terug in het CX register geplaatst.

→ POP CX



Na de POP instructie;

- Het register is terug gevuld met de waard van de laatste pop instructie. (LIFO !!)

Push and Pop

PUSH/POP met trace:

```
-a 100
1469:0100 MOV AX, AA01
1469:0103 MOV BX, BB01
1469:0106 MOV CX, CC01
1469:0109 MOV DX, DD01
1469:010C PUSH AX
1469:010D PUSH BX
1469:010E PUSH CX,
1469:010F PUSH DX
1469:0110 MOV AX, AA02
1469:0113 MOV BX, BB02
1469:0116 MOV CX, CC02
1469:0119 MOV DX, DD02
1469:011C POP DX
1469:011D POP CX
1469:011E POP BX
1469:011F POP AX
1469:0120
```

Voer de code in assembler.

Controleer de werking van push and pop met het trace commando.

Push and Pop

Voorbeeld oefening: PUSH AND POP

- Vul een scherm met ♣ (♣ = ASCII 05).
- 40 tekens per rij
- 20 rijen onder elkaar
- Maak gebruik van INT21 met subfunctie 02 .
- Maak geen gebruik van een hulpregister, maar gebruik push & pop.

[illegible]

Push and Pop

Voorbeeld oefening: PUSH AND POP

CX register vullen

Waarde van CX naar STACK

1481:0100	B402	MOV	AH, 02
1481:0102	B91400	MOV	CX, 0014
1481:0105	51	PUSH	CX
1481:0106	B92800	MOV	CX, 0028
1481:0109	B205	MOV	DL, 05
1481:010B	CD21	INT	21
1481:010D	E2FC	LOOP	010B
1481:010F	B20A	MOV	DL, 0A
1481:0111	CD21	INT	21
1481:0113	B20D	MOV	DL, 0D
1481:0115	CD21	INT	21
1481:0117	59	POP	CX
1481:0118	E2EB	LOOP	0105
1481:011A	CD20	INT	20

Push and Pop

Voorbeeld oefening: PUSH AND POP

Printen van karakter (ASCII 05)
40 x na elkaar (LOOP)

```
1481:0100 B402      MOV     AH, 02
1481:0102 B91400     MOV     CX, 0014
1481:0105 51        PUSH    CX
1481:0106 B92800     MOV     CX, 0028
1481:0109 B205      MOV     DL, 05
1481:010B CD21      INT     21
1481:010D E2FC      LOOP    010B
1481:010F B20A      MOV     DL, 0A
1481:0111 CD21      INT     21
1481:0113 B20D      MOV     DL, 0D
1481:0115 CD21      INT     21
1481:0117 59        POP     CX
1481:0118 E2EB      LOOP    0105
1481:011A CD20      INT     20
```


Push and Pop

Voorbeeld oefening: PUSH AND POP

```
1481:0100 B402      MOV     AH,02
1481:0102 B91400     MOV     CX,0014
1481:0105 51        PUSH    CX
1481:0106 B92800     MOV     CX,0028
1481:0109 B205      MOV     DL,05
1481:010B CD21      INT     21
1481:010D E2FC      LOOP    010B
1481:010F B20A      MOV     DL,0A
1481:0111 CD21      INT     21
1481:0113 B20D      MOV     DL,0D
1481:0115 CD21      INT     21
1481:0117 59        POP     CX
1481:0118 E2EB      LOOP    0105
1481:011A CD20      INT     20
```

Printen van één enter

Push and Pop

Voorbeeld oefening: PUSH AND POP

```
1481:0100 B402      MOV     AH, 02
1481:0102 B91400    MOV     CX, 0014
1481:0105 51        PUSH    CX
1481:0106 B92800    MOV     CX, 0028
1481:0109 B205      MOV     DL, 05
1481:010B CD21      INT     21
1481:010D E2FC      LOOP    010B
1481:010F B20A      MOV     DL, 0A
1481:0111 CD21      INT     21
1481:0113 B20D      MOV     DL, 0D
1481:0115 CD21      INT     21
1481:0117 59        POP     CX
1481:0118 E2EB      LOOP    0105
1481:011A CD20      INT     20
```

*De loop werkt met de CX waarde die in de vorige instructie van stack is gehaald (POP CX). De Loop instructie verminderd de CX waarde met 1. Is deze waarde niet gelijk aan nul, sprint het programma naar IP 105.
Hier wordt de (nieuwe) CX waarde terug op stack geplaatst.*

Push and Pop

Voorbeeld oefening: PUSH AND POP

```
1481:0100 B402      MOV     AH, 02
1481:0102 B91400    MOV     CX, 0014
1481:0105 51        PUSH    CX
1481:0106 B92800    MOV     CX, 0028
1481:0109 B205      MOV     DL, 05
1481:010B CD21      INT     21
1481:010D E2FC      LOOP    010B
1481:010F B20A      MOV     DL, 0A
1481:0111 CD21      INT     21
1481:0113 B20D      MOV     DL, 0D
1481:0115 CD21      INT     21
1481:0117 59        POP     CX
1481:0118 E2EB      LOOP    0105
1481:011A CD20      INT     20
```

De loop instructie werkt met de CX waarde die in de vorige instructie van stack is gehaald (POP CX). De Loop instructie vermindert de CX waarde met 1. Is deze waarde niet gelijk aan nul, sprint het programma naar IP 105. Hier wordt de (nieuwe) CX waarde terug op stack geplaatst.