# ISA Multiple Data Transfer (3A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

# Based on

ARM System-on-Chip Architecture, 2nd ed, Steve Furber

# Stack Types and Stack Top Operations

(F,E) x (A,D) = { FA, FD, EA, ED }

**Stack Types** – Semantics

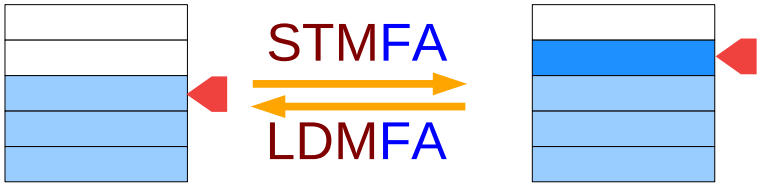
PUSH(STM) / POP(LDM) on an { FA / FD / EA / ED } type stack

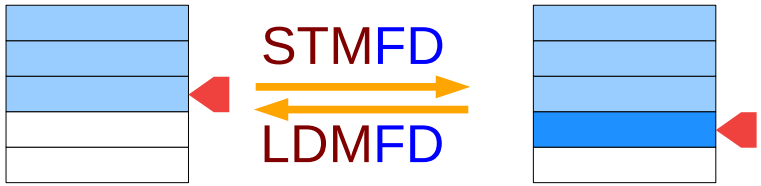
(I,D) x (B,A) = { IB, IA, DB, DA }

**Stack Top Operations** – Syntax

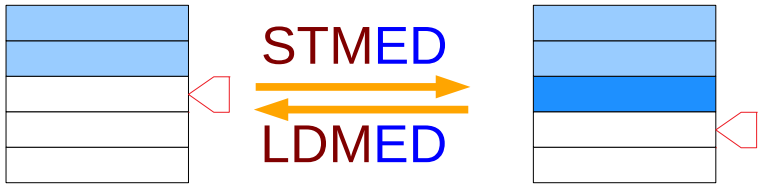
Do { Inc / Dec } stack top operation { Before / After } STM / LDM

# Inverse Stack Operations



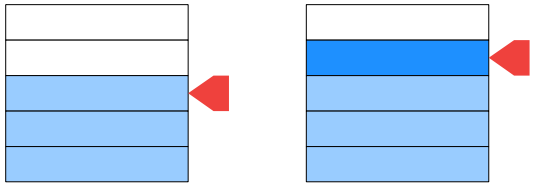





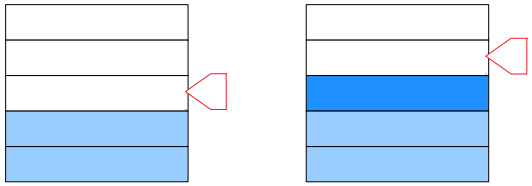


# Complementary Stack Types

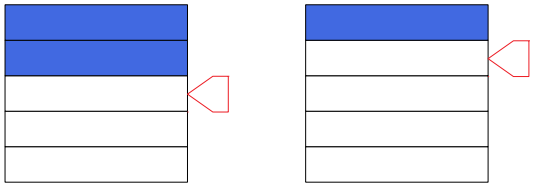
STMIB
FA stack



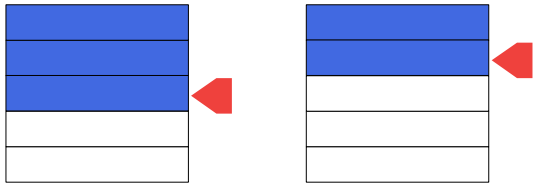
STMIA
EA stack



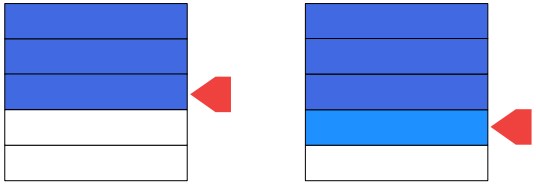
LDMIB
ED stack



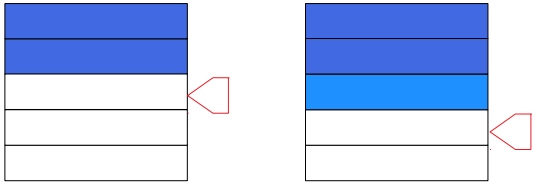
LDMIA
FD stack



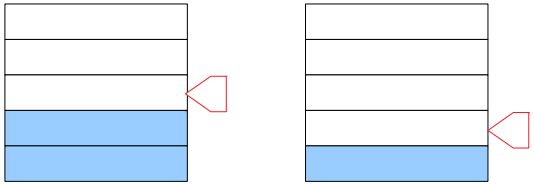
STMDB
FD stack



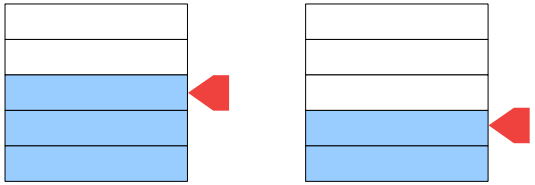
STMDA
ED stack



LDMDB
EA stack



LDMDA
FA stack



# Inverse Stack Operations and Complementary Stacks

STMFA
LDMFA

STMFD
LDMFD

STMEA
LDMEA

STMED
LDMED

**Inverse Stack Operations**

STMIB
LDMIB

STMIA
LDMIA

STMDB
LDMDB

STMDA
LDMDA

**Complementary Stacks**

# (Empty / Full) and (Before / After)

| | | | | | | |
|---|---|---|---|---|---|---|
| STM | occupied<br>Full | A/D | = | STM | I/D | Inc / dec before<br>Before |
| STM | empty<br>Empty | A/D | = | STM | I/D | Inc / dec later<br>After |
| LDM | occupied<br>Full | A/D | ≠ | LDM | I/D | Inc / dec before<br>Before |
| LDM | empty<br>Empty | A/D | | LDM | I/D | Inc / dec later<br>After |

□ = { Ascend, Descend }

□ = { Inc, Dec }

# (F_ / E_) and (_B / _A) reasoning

| | | | |
|---|---|---|---|
| STMF□ | If the stack top is **full** | STM□B | then inc / dec the stack pointer **before** storing a new element |
| STME□ | If the stack top is **empty** | STM□A | then inc / dec the stack pointer **after** storing a new element |
| LDMF□ | If the stack top is **full** | LDM□A | then inc / dec the stack pointer **after** getting an element |
| LDME□ | If the stack top is **empty** | LDM□B | then inc / dec the stack pointer **before** getting an element |

□ = { **A**scend, **D**escend }

□ = { **I**nc, **D**ec }

# (Ascend / Descend) and (Increment / Decrement)

| | | | | | | |
|---|---|---|---|---|---|---|
| STM | F/E | PUSH<br>Ascend | = | STM | Inc | the same direction<br>B/A |
| STM | F/E | PUSH<br>Descend | = | STM | Dec | the same direction<br>B/A |
| LDM | F/E | POP<br>Ascend | ≠ | LDM | Inc | the opposite direction<br>B/A |
| LDM | F/E | POP<br>Descend | ≠ | LDM | Dec | the opposite direction<br>B/A |

□ = { **F**ull / **E**mpty }

□ = { **B**efore / **A**fter }

# (_A / _D) and (I_ / D_) reasoning

| | | | |
|---|---|---|---|
| STM□A | To push onto the **ascending** stack | STMI□ | **Increment** the stack top pointer |
| STM□D | To push onto the **descending** stack | STMD□ | **Decrement** the stack top pointer |
| LDM□A | To pop from the **ascending** stack | LDMD□ | **Decrement** the stack top pointer |
| LDM□D | To pop from the **descending** stack | LDMI□ | **Increment** the stack top pointer |

□ = { **F**ull / **E**mpty }

□ = { **B**efore / **A**fter }

# STM / LDM Equivalence Summary

## STM

occupied — Full — Inc / dec before — Before

empty — Empty — Inc / dec later — After

## STM

PUSH — Ascend — the same direction — Inc

PUSH — Descend — the same direction — Dec

## LDM

occupied — Full — Inc / dec before — Before

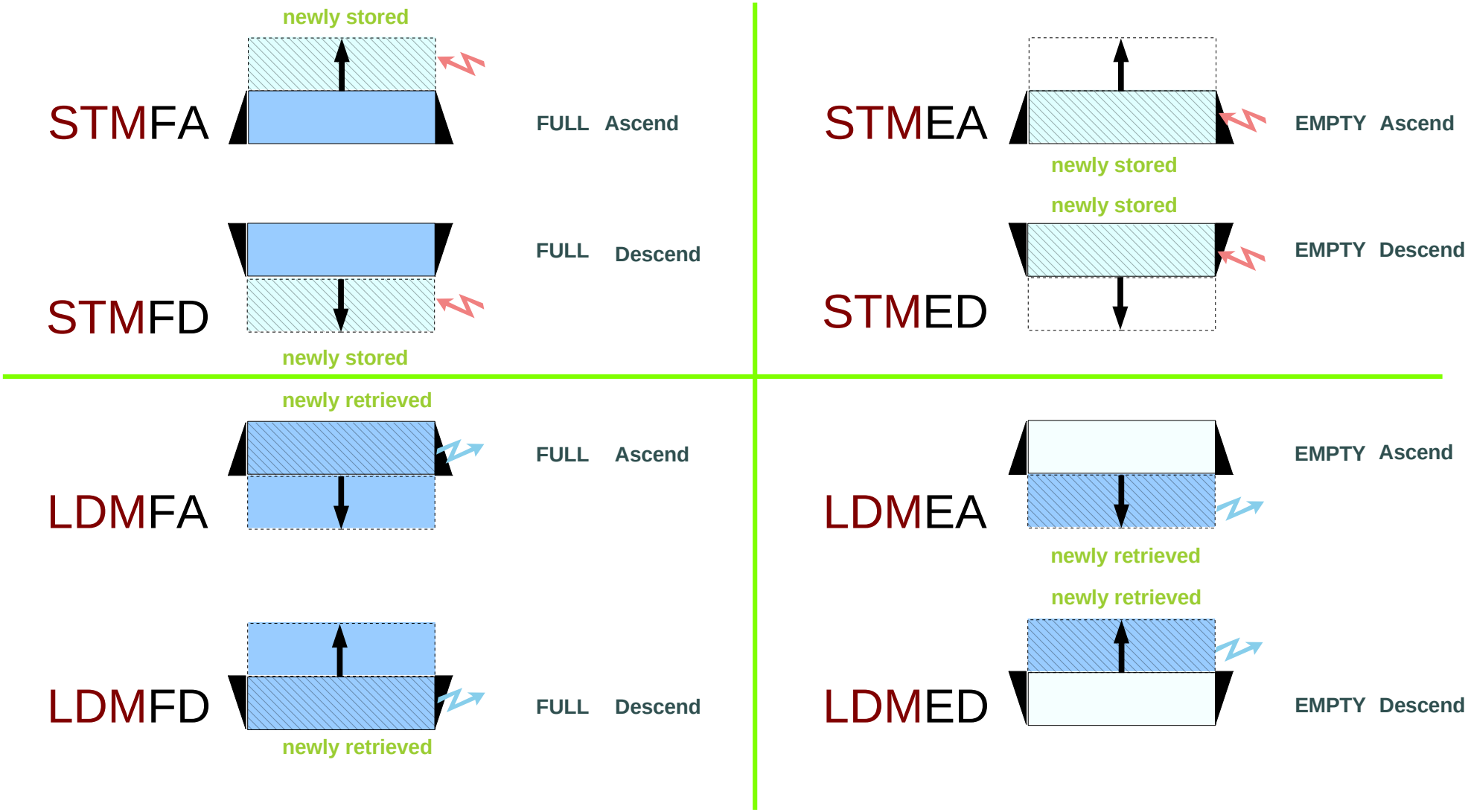
empty — Empty — Inc / dec later — After

## LDM

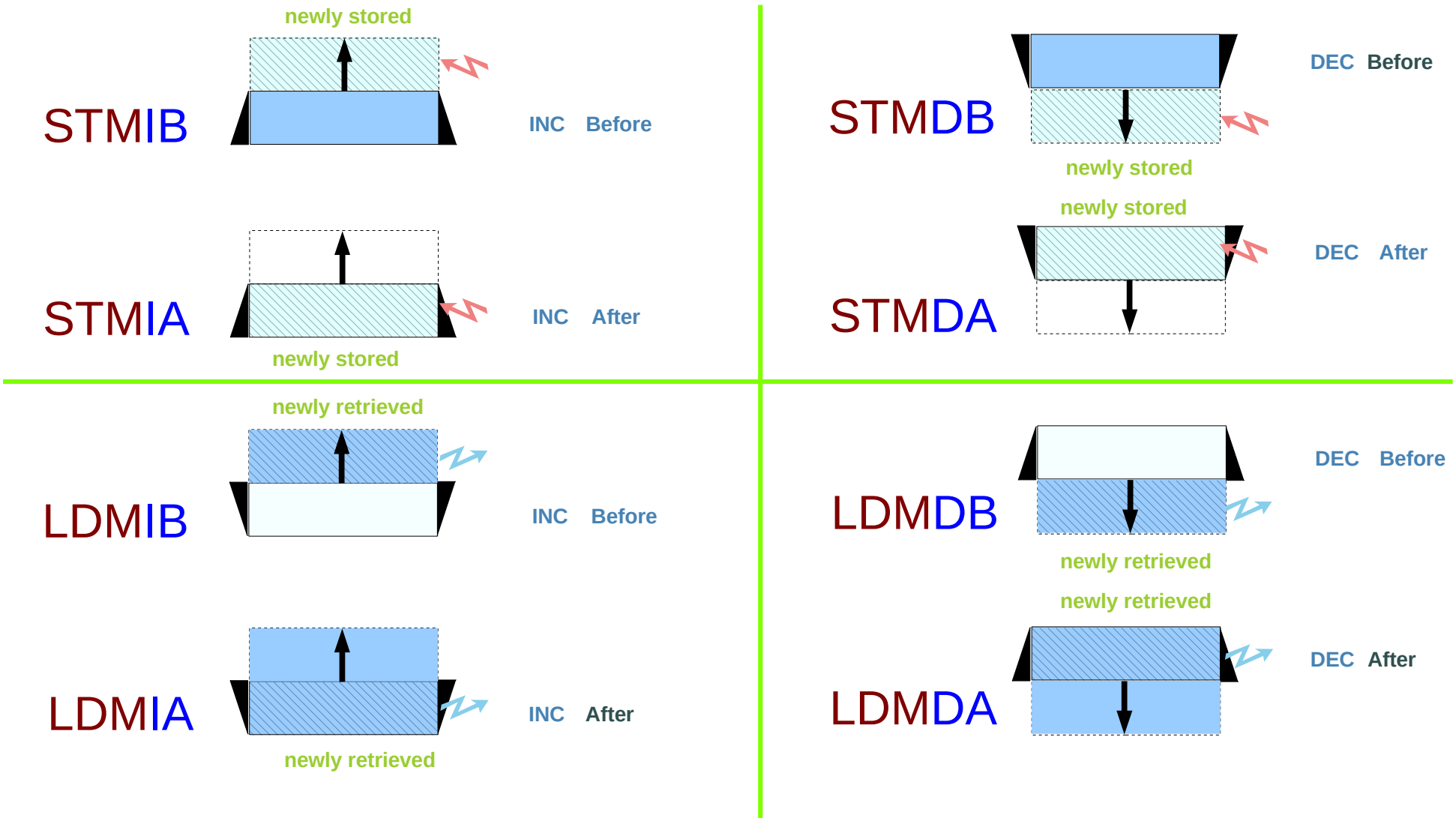
POP — Ascend — the opposite direction — Inc

POP — Descend — the opposite direction — Dec

# Stack View – (STM,LDM) x (F,E) x (A,D)



# Block Copy View – (STM,LDM) x (I, D) x (B,A)



# Block copy view ➡ Stack view

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| STMIB ↑ | PUSH | ↑ | Ascending | Inc Before ST | | Full | STMFA ↑ |
| STMIA ↑ | PUSH | ↑ | Ascending | Inc After ST | | Empty | STMEA ↑ |
| STMDB ↓ | PUSH | ↓ | Descending | Dec Before ST | | Full | STMFD ↓ |
| STMDA ↓ | PUSH | ↓ | Descending | Dec After ST | | Empty | STMED ↓ |
| LDMIB ↑ | POP | ↓ | Descending | Inc Before LD | | Empty | LDMED ↓ |
| LDMIA ↑ | POP | ↓ | Descending | Inc After LD | | Full | LDMFD ↓ |
| LDMDB ↓ | POP | ↑ | Ascending | Dec Before LD | | Empty | LDMEA ↑ |
| LDMDA ↓ | POP | ↑ | Ascending | Dec Before LD | | Full | LDMFA ↑ |

# Stack view ➡ Block copy view

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| STMFA | Full | Ascending | PUSH | Inc Before ST | STMIB |
| STMEA | Empty | Ascending | PUSH | Inc After ST | STMIA |
| STMFD | Full | Descending | PUSH | Dec Before ST | STMDB |
| STMED | Empty | Descending | PUSH | Dec After ST | STMDA |
| LDMED | Empty | Descending | POP | Inc Before LD | LDMIB |
| LDMFD | Full | Descending | POP | Inc After LD | LDMIA |
| LDMEA | Empty | Ascending | POP | Dec Before LD | LDMDB |
| LDMFA | Full | Ascending | POP | Dec Before LD | LDMDA |

# STM Equivalent Operations – (I,D) x (B,A)

STM

Full Ascending

Increasing Before ST

**IB = FA**

STM

Empty Ascending

Increasing After ST

**IA = EA**

STM

Full Descending

Decreasing Before ST

**DB = FD**

STM

Empty Descending

Decreasing After ST

**DA = ED**

# LDM Equivalent Operations – (I,D) x (B,A)

LDM

Empty Descending

Increasing Before LD

**IB = ED**

LDM

Full Descending

Increasing After LD

**IA = FD**

LDM

Empty Ascending

Decreasing Before LD

**DB = EA**

LDM

Full Ascending

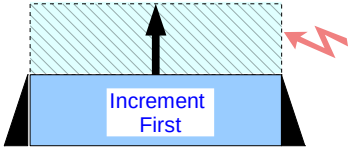
Decreasing After LD

**DA = FA**

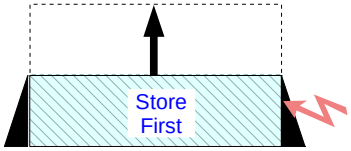
# (STM, LDM) x (I, D) x (B, A) orders

## Ascending Stack PUSH

STMIB




STMIA

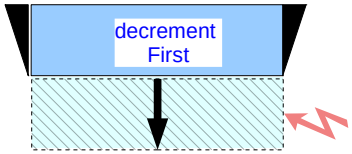



STMFA

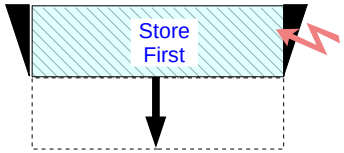
STMEA

## Descending Stack PUSH

STMDB




STMDA

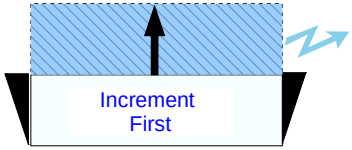



STMFD

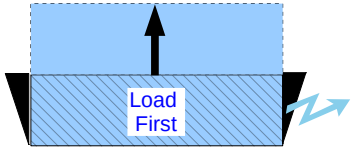
STMED

## Descending Stack POP

LDMIB




LDMIA

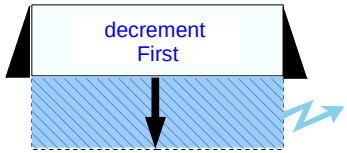



LDMED

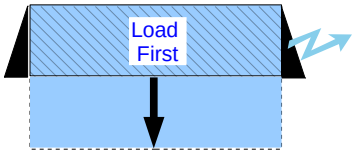
LDMFD

## Ascending Stack POP

LDMDB




LDMDA



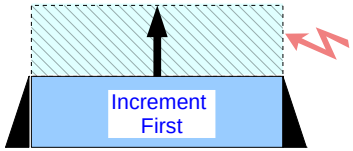

LDMEA

LDMFA

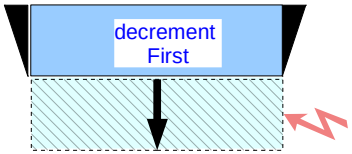
# (STM, LDM) x (F, E) x (A, D) orders

## Full Top PUSH

STMFA




STMFD

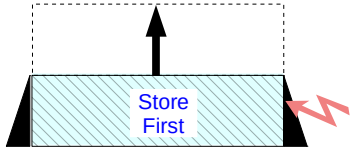



STMIB

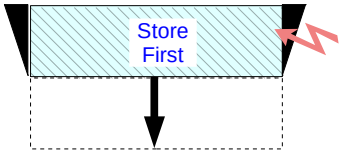
STMDB

## Empty Top PUSH

STMEA




STMED

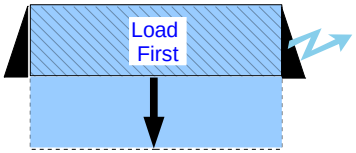



STMIA

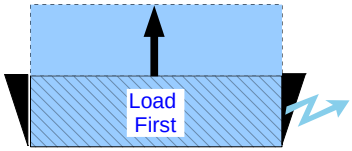
STMDA

## Full Top POP

LDMFA




LDMFD

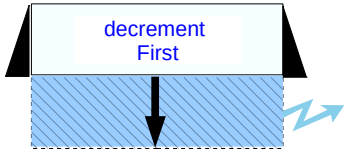



LDMDA

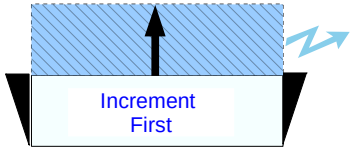
LDMIA

## Empty Top POP

LDMEA




LDMED




LDMDB

LDMIB

# (STM, LDM) x (I, D) x (B, A) orders

## Ascending Stack PUSH

| | increment before | increment after |
|---|---|---|
| STM | I B | I A |
| STM | F A | E A |
| | full ascending | empty ascending |

## Descending Stack PUSH

| | decrement before | decrement after |
|---|---|---|
| STM | D B | D A |
| STM | F D | E D |
| | full descending | empty descending |

## Descending Stack POP

| | increment before | increment after |
|---|---|---|
| LDM | I B | I A |
| LDM | E D | F D |
| | empty descending | full descending |

## Ascending Stack POP

| | decrement before | decrement after |
|---|---|---|
| LDM | D B | D A |
| LDM | E A | F A |
| | empty ascending | full ascending |

# (STM, LDM) x (E, F) x (A, D) orders

## Empty Top PUSH

| | increment | after |
|---|---|---|
| STM | I | A |
| STM | E | A |
| | empty | ascending |

| | decrement | after |
|---|---|---|
| | D | A |
| | E | D |
| | empty | descending |

## Full Top PUSH

| | increment | before |
|---|---|---|
| STM | I | B |
| STM | F | A |
| | full | ascending |

| | decrement | before |
|---|---|---|
| | D | B |
| | F | D |
| | full | descending |

## Empty Top POP

| | decrement | before |
|---|---|---|
| LDM | D | B |
| LDM | E | A |
| | empty | ascending |

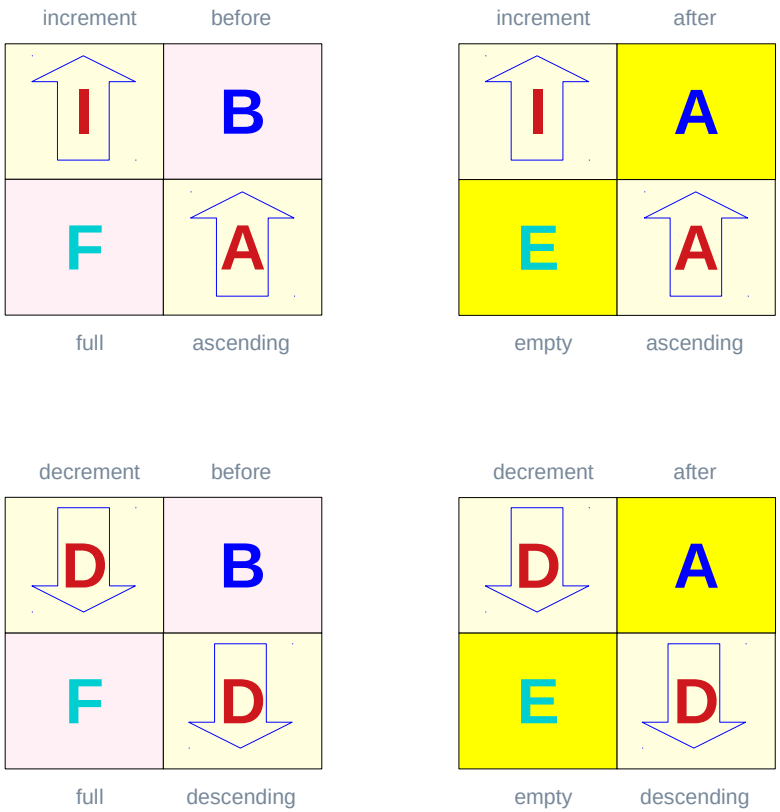
| | increment | before |
|---|---|---|
| | I | B |
| | E | D |
| | empty | descending |

## Full Top POP

| | decrement | after |
|---|---|---|
| LDM | D | A |
| LDM | F | A |
| | full | ascending |

| | increment | after |
|---|---|---|
| | I | A |
| | F | D |
| | full | descending |

# **STM** in the <u>same</u> direction : (Inc – Asc), (Dec – Dsc)

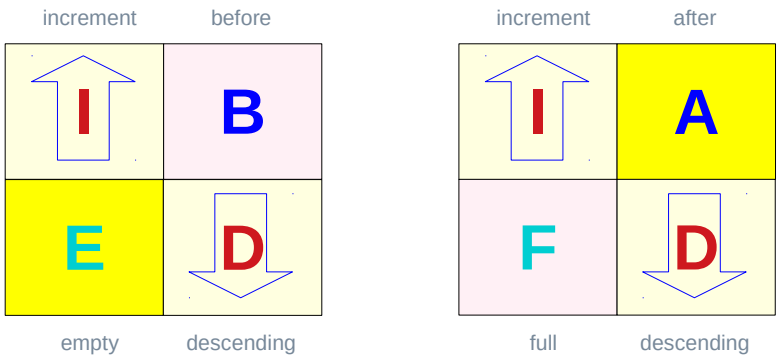



**Ascending**

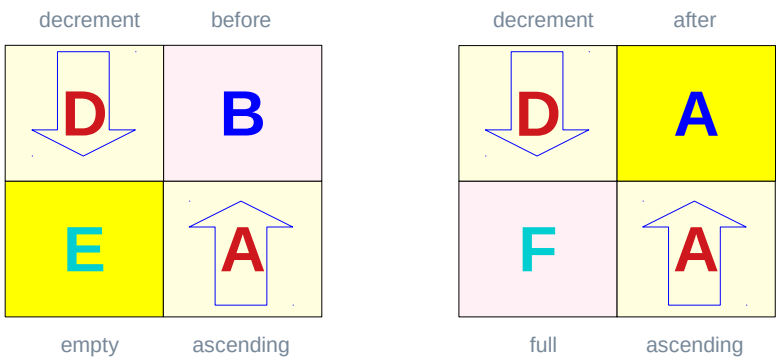
**Descending**

**FULL** **EMPTY**

# **LDM** in the opposite direction : (Inc – Dsc), (Dec – Asc)




**Descending**

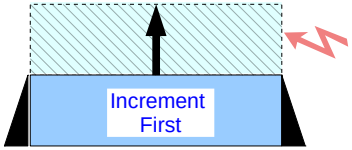



**Ascending**

**EMPTY** **FULL**

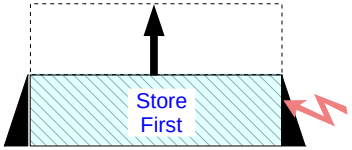
# Equivalence – (STM, LDM) x (I, D) x (B, A)

STMIB



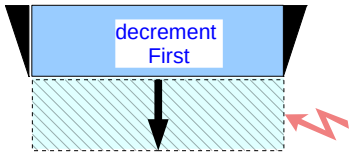

STMFA

STMIA



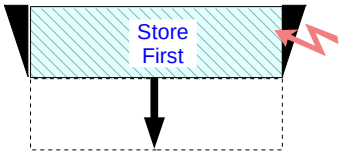

STMEA

STMDB



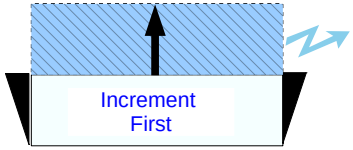

STMFD

STMDA



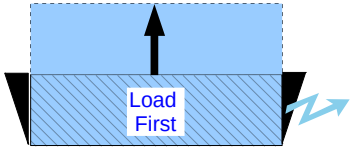

STMED

LDMIB



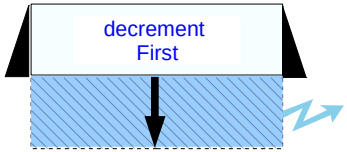

LDMED

LDMIA



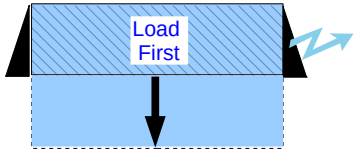

LDMFD

LDMDB




LDMEA

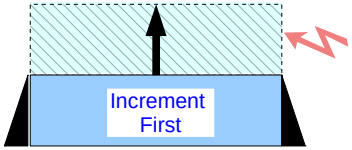
LDMDA




LDMFA

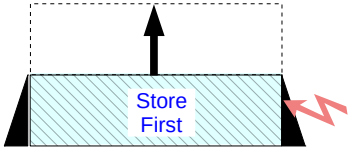
# Equivalence – (STM, LDM) x (F, E) x (A, D)

STMIB



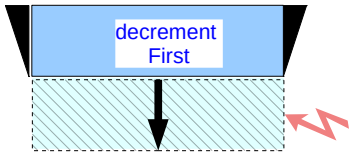

STMFA

STMIA



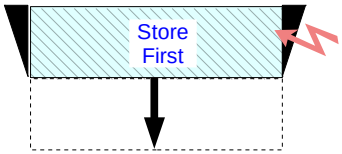

STMEA

STMDB



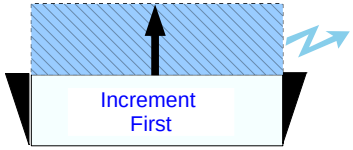

STMFD

STMDA



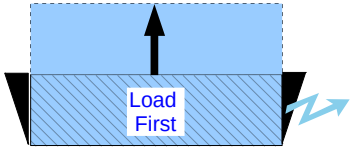

STMED

LDMDA



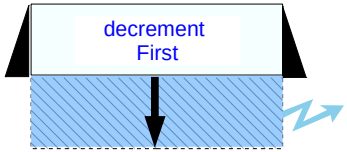

LDMFA

LDMDB



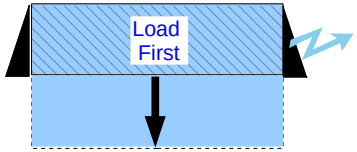

LDMEA

LDMIA



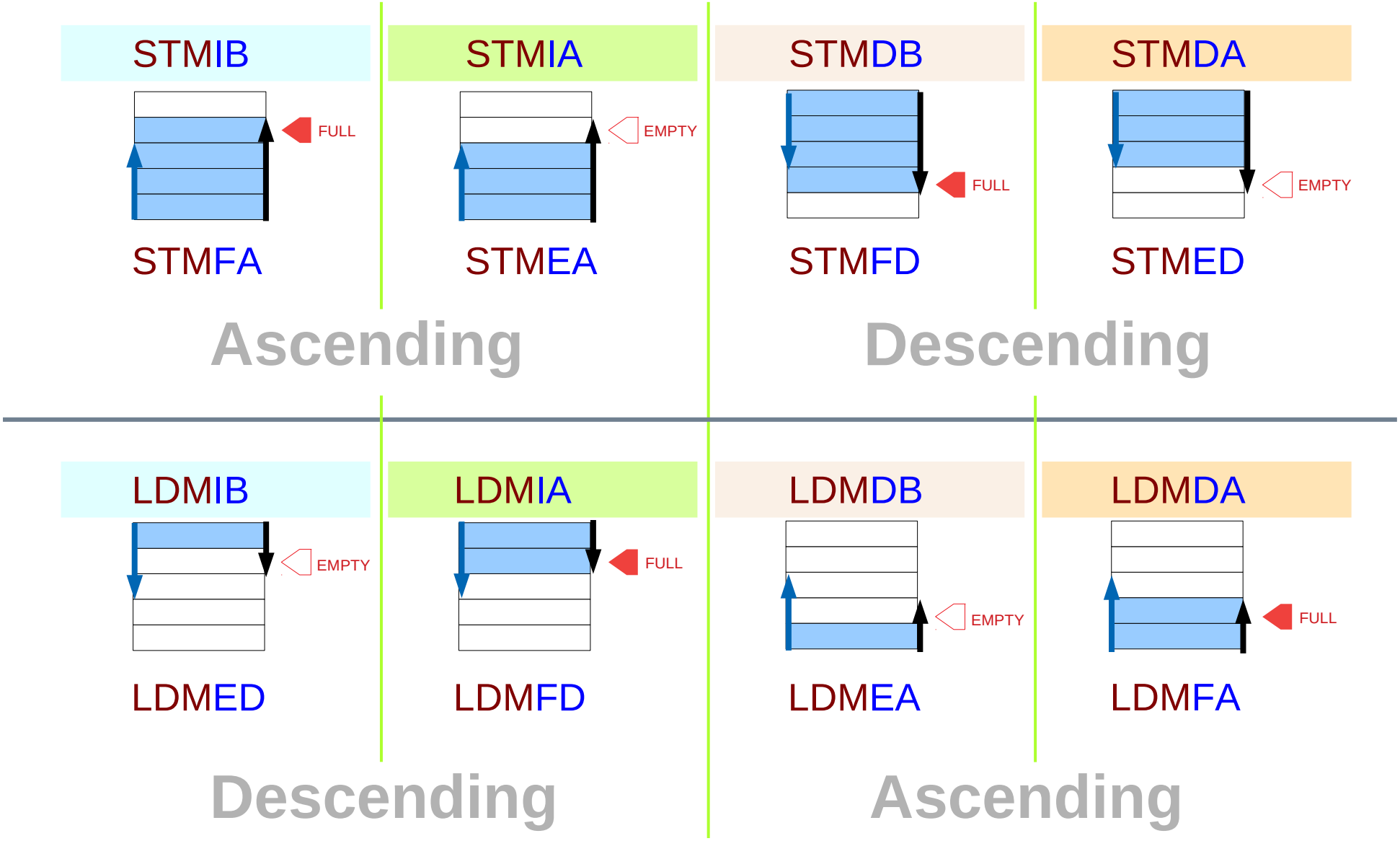

LDMFD

LDMIB



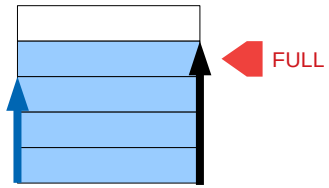

LDMED

# Stack view – (STM, LDM) x (I, D) x (B, A)



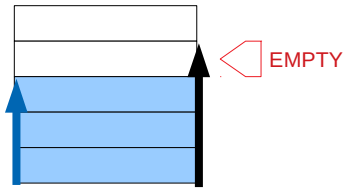
# Stack view – (STM, LDM) x (F, E) x (A, D)

STMIB



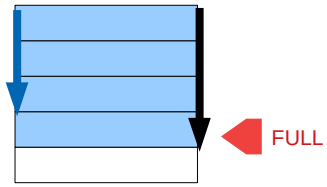

STMFA

STMIA



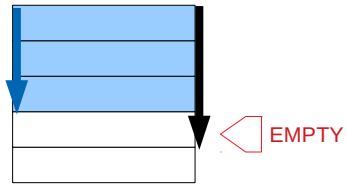

STMEA

STMDB




STMFD

STMDA

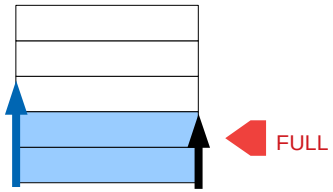



STMED

**Ascending**

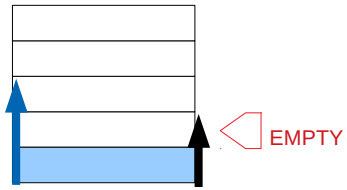
**Descending**

LDMDA



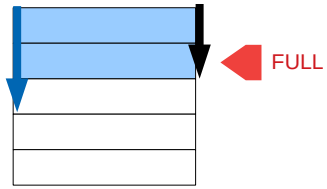

LDMFA

LDMDB



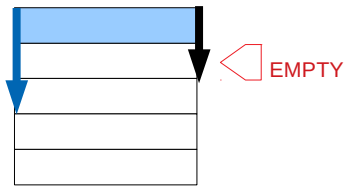

LDMEA

LDMIA




LDMFD

LDMIB

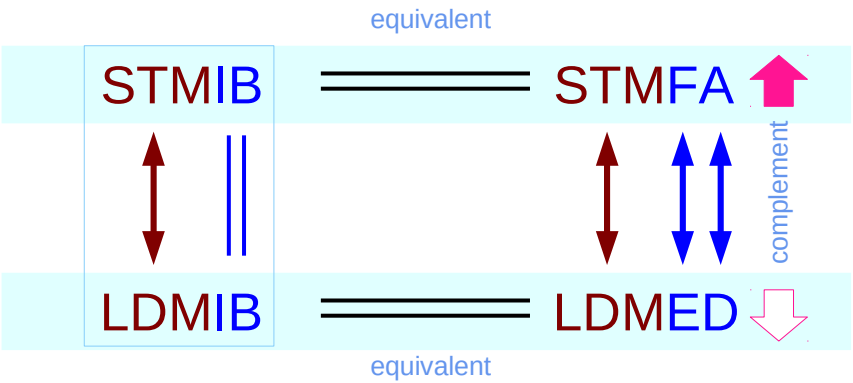



LDMED

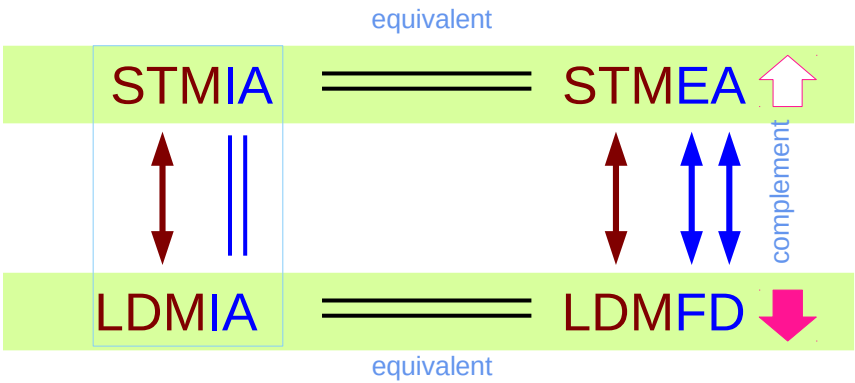
**Ascending**

**Descending**

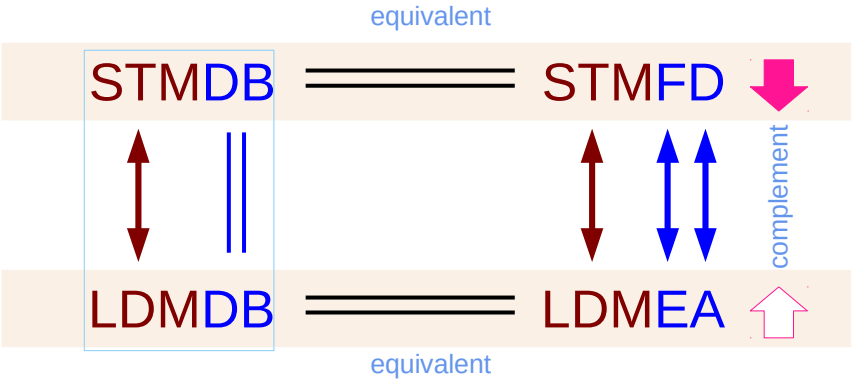
# Equivalent & Complementary Relations – (I,D) x(B, A)



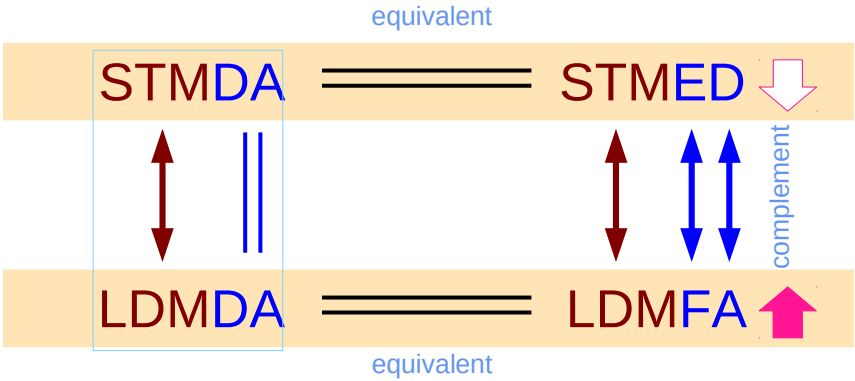

**Complementary Stacks**




**Complementary Stacks**

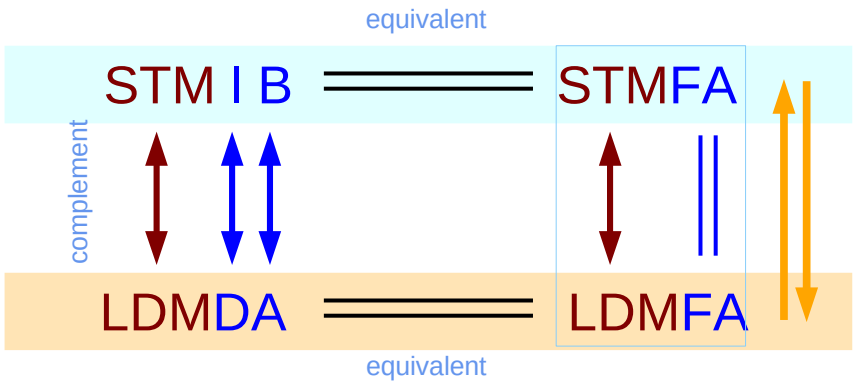



**Complementary Stacks**

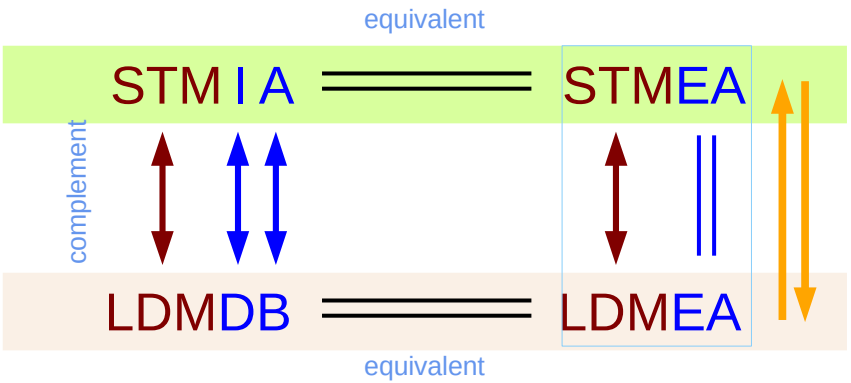



**Complementary Stacks**

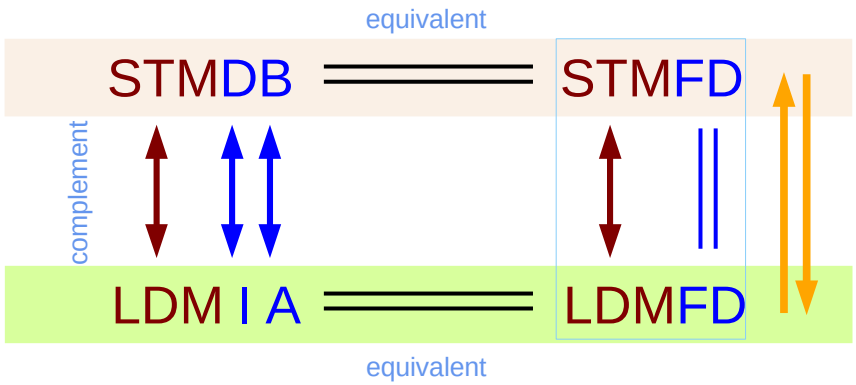
# Equivalent & Complementary Relations – (F,E) x (A,D)



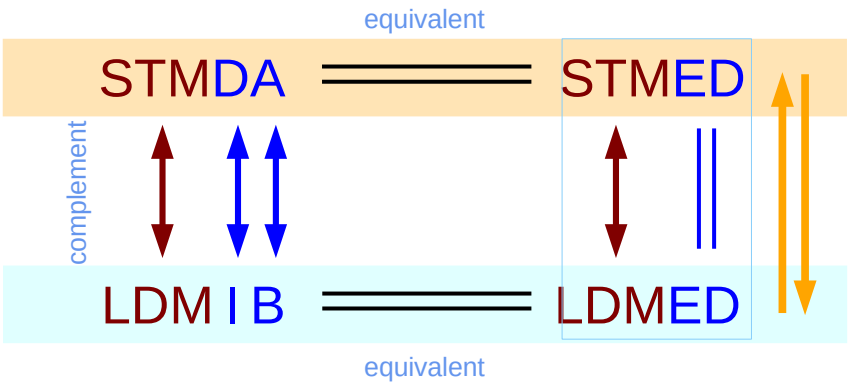

**Inverse Stack Operations**




**Inverse Stack Operations**




**Inverse Stack Operations**




**Inverse Stack Operations**

# Examples of Multiple Data Transfers

# Multiple Data Transfer – STM (I,D)x(B,A)

## STM

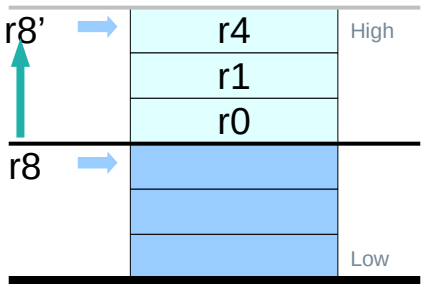
! auto index

**Full Ascending**

**Increment Before ST**

PUSH

first, increment; then, store




```
STMIB r8! {r0,r1,r4}
STMFA r8! {r0,r1,r4}
```

## STM

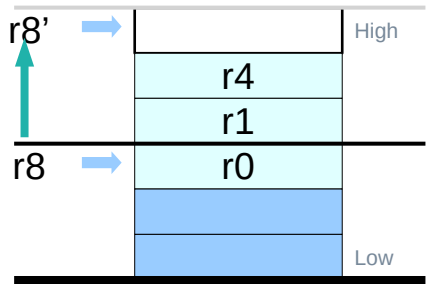
! auto index

**Empty Ascending**

**Increment After ST**

PUSH

first, store; then, increment




```
STMIA r8! {r0,r1,r4}
STMEA r8! {r0,r1,r4}
```

## STM

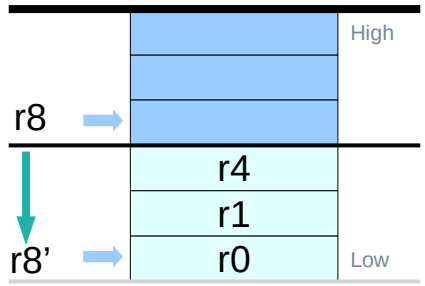
! auto index

**Full Descending**

**Decrement Before ST**

first, decrement; then, store

PUSH




```
STMDB r8! {r0,r1,r4}
STMFD r8! {r0,r1,r4}
```

## STM

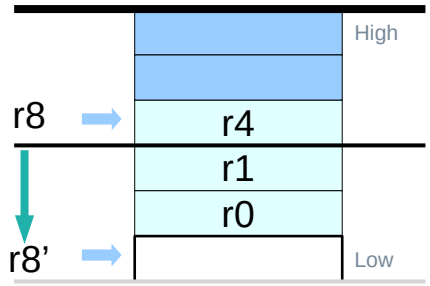
! auto index

**Empty Descending**

**Decrement After ST**

first, store; then, decrement

PUSH




```
STMDA r8! {r0,r1,r4}
STMED r8! {r0,r1,r4}
```

# Multiple Data Transfer – LDM (I,D)x(B,A)

## LDM

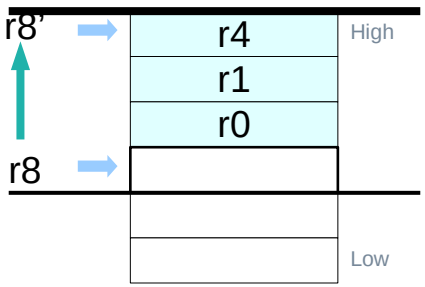
! auto index

**Empty Descending**

**Increment Before LD**

POP

first, increment; then, store




```
LDMIB r8! {r0,r1,r4}
LDMED r8! {r0,r1,r4}
```

## LDM

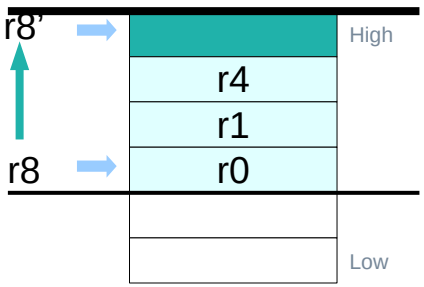
! auto index

**Full Descending**

**Increment After LD**

POP

first, store; then, increment




```
LDMIA r8! {r0,r1,r4}
LDMFD r8! {r0,r1,r4}
```

## LDM

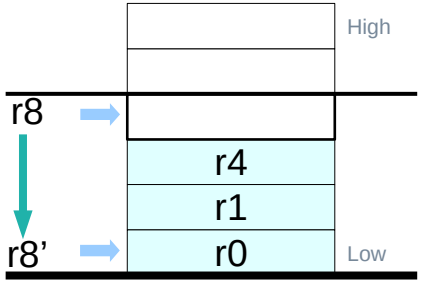
! auto index

**Empty Ascending**

**Decrement Before LD**

first, decrement; then, store

POP




```
LDMDB r8! {r0,r1,r4}
LDMEA r8! {r0,r1,r4}
```

## LDM

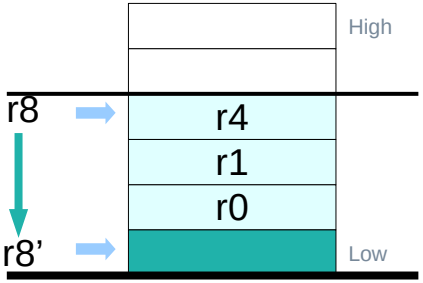
! auto index

**Full Ascending**

**Decrement After LD**

first, store; then, decrement

POP




```
LDMDA r8! {r0,r1,r4}
LDMFA r8! {r0,r1,r4}
```

# A Multiple Data Transfer
## as a sequence of Single Data Transfers

# STM with auto-indexing – (I,D) x (B,A)

```
STMIB   r8!,   ⬅ {r0,r1,r4}
STMED   r8!,   ⬅ {r0,r1,r4}

STR     r0,    ➡ [r8, #4]!
STR     r1,    ➡ [r8, #4]!
STR     r4,    ➡ [r8, #4]!
```

```
STMIA   r8!,   ⬅ {r0,r1,r4}
STMFD   r8!,   ⬅ {r0,r1,r4}

STR     r0,    ➡ [r8], #1
STR     r1,    ➡ [r8], #1
STR     r4,    ➡ [r8], #1
```

```
STMDB   r0!,   ⬅ {r0,r1,r4}
STMEA   r0!,   ⬅ {r0,r1,r4}

STR     r4,    ➡ [r0, #-4]!
STR     r1,    ➡ [r0, #-4]!
STR     r0,    ➡ [r0, #-4]!
```

```
STMDA   r8!,   ⬅ {r0,r1,r4}
STMFA   r8!,   ⬅ {r0,r1,r4}

STR     r4,    ➡ [r8], #-4
STR     r1,    ➡ [r8], #-4
STR     r0,    ➡ [r8], #-4
```

# **LDM** with auto-indexing – (I,D) x (B, A)

```
LDMIB   r8!,   ➡ {r0,r1,r4}
LDMED   r8!,   ➡ {r0,r1,r4}

LDR     r0,    ⬅ [r8, #4]!
LDR     r1,    ⬅ [r8, #4]!
LDR     r4,    ⬅ [r8, #4]!
```

```
LDMIA   r8!,   ➡ {r0,r1,r4}
LDMFD   r8!,   ➡ {r0,r1,r4}

LDR     r0,    ⬅ [r8], #4
LDR     r1,    ⬅ [r8], #4
LDR     r4,    ⬅ [r8], #4
```

```
LDMDB   r8!,   ➡ {r0,r1,r4}
LDMEA   r8!,   ➡ {r0,r1,r4}

LDR     r4,    ⬅ [r8, #-4]!
LDR     r1,    ⬅ [r8, #-4]!
LDR     r0,    ⬅ [r8, #-4]!
```

```
LDMDA   r8!,   ➡ {r0,r1,r4}
LDMFA   r8!,   ➡ {r0,r1,r4}

LDR     r4,    ⬅ [r8], #-4
LDR     r1,    ⬅ [r8], #-4
LDR     r0,    ⬅ [r8], #-4
```

# STM without auto-indexing – (I,D) x (B,A)

```
STMIB   r8,   ⬅ {r0,r1,r4}
STMED   r8,   ⬅ {r0,r1,r4}

STR     r0,   ➡ [r8, #4]
STR     r1,   ➡ [r8, #8]
STR     r4,   ➡ [r8, #12]
```

```
STMIA   r8,   ⬅ {r0,r1,r4}
STMFD   r8,   ⬅ {r0,r1,r4}

STR     r0,   ➡ [r8, #0]
STR     r1,   ➡ [r8, #4]
STR     r4,   ➡ [r8, #8]
```

```
STMDB   r8,   ⬅ {r0,r1,r4}
STMEA   r8,   ⬅ {r0,r1,r4}

STR     r4,   ➡ [r0, #-4]
STR     r1,   ➡ [r0, #-8]
STR     r0,   ➡ [r0, #-12]
```

```
STMDA   r8,   ⬅ {r0,r1,r4}
STMFA   r8,   ⬅ {r0,r1,r4}

STR     r4,   ➡ [r8, #0]
STR     r1,   ➡ [r8, #-4]
STR     r0,   ➡ [r8, #-8]
```

# LDM without auto-indexing – (I,D) x (B,A)

```
LDMIB   r8,   ➡ {r0,r1,r4}
LDMED   r8,   ➡ {r0,r1,r4}

LDR     r0,   ⬅ [r8, #4]
LDR     r1,   ⬅ [r8, #8]
LDR     r4,   ⬅ [r8, #12]
```

```
LDMIA   r8,   ➡ {r0,r1,r4}
LDMFD   r8,   ➡ {r0,r1,r4}

LDR     r2,   ⬅ [r8, #0]
LDR     r3,   ⬅ [r8, #4]
LDR     r4,   ⬅ [r8, #8]
```

```
LDMDB   r8,   ➡ {r0,r1,r4}
LDMEA   r8,   ➡ {r0,r1,r4}

LDR     r4,   ⬅ [r8, #-4]
LDR     r1,   ⬅ [r8, #-8]
LDR     r0,   ⬅ [r8, #-12]
```

```
LDMDA   r8,   ➡ {r0,r1,r4}
LDMFA   r8,   ➡ {r0,r1,r4}

LDR     r4,   ⬅ [r8, #0]
LDR     r1,   ⬅ [r8, #-4]
LDR     r0,   ⬅ [r8, #-6]
```

A Multiple Data Transfer

- conceptual
  - increment / decrement
  - before / after
  - STM / LDM

# Multiple Data Transfer – STM (I,D)x(B,A)

## STM

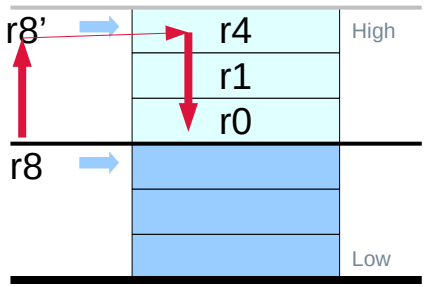
! auto index

**Full Ascending**

**Increment Before ST**

PUSH

first, increment; then, store




```
STMIB r8! {r0,r1,r4}
STMFA r8! {r0,r1,r4}
```

Before

## STM

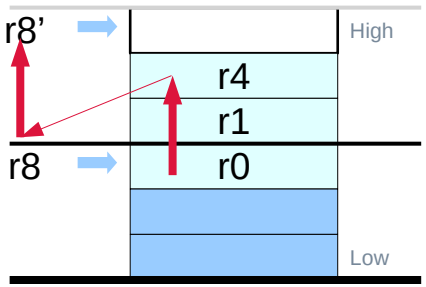
! auto index

**Empty Ascending**

**Increment After ST**

PUSH

first, store; then, increment




```
STMIA r8! {r0,r1,r4}
STMEA r8! {r0,r1,r4}
```

After

## STM

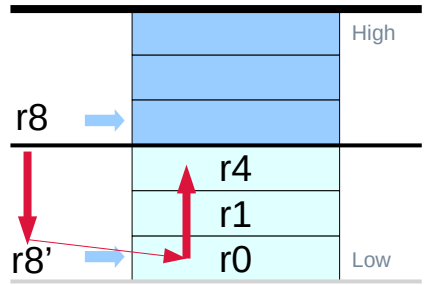
! auto index

**Full Descending**

**Decrement Before ST**

first, decrement; then, store

PUSH




```
STMDB r8! {r0,r1,r4}
STMFD r8! {r0,r1,r4}
```

Before

## STM

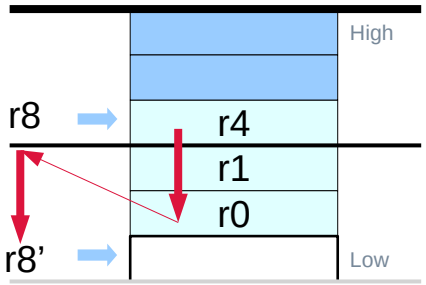
! auto index

**Empty Descending**

**Decrement After ST**

first, store; then, decrement

PUSH




```
STMDA r8! {r0,r1,r4}
STMED r8! {r0,r1,r4}
```

After

# Multiple Data Transfer – LDM (I,D)x(B,A)

## Before

### LDM

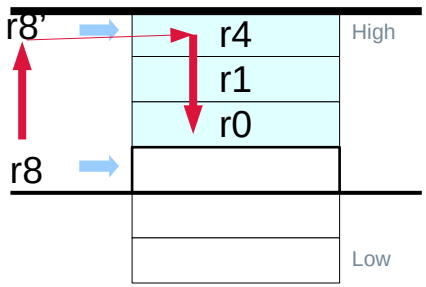
! auto index

**Increment Before LD**

POP

first, increment; then, store

**Empty Descending**




```
LDMIB r8! {r0,r1,r4}
LDMED r8! {r0,r1,r4}
```

### LDM

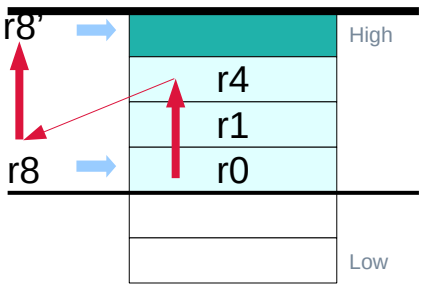
! auto index

**Increment After LD**

POP

first, store; then, increment

**Full Descending**




```
LDMIA r8! {r0,r1,r4}
LDMFD r8! {r0,r1,r4}
```

## Before

### LDM

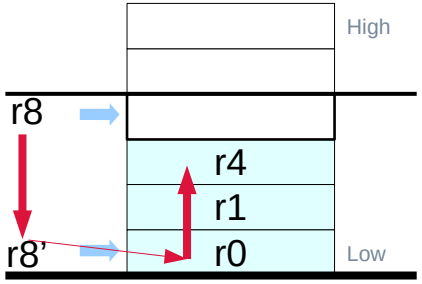
! auto index

**Decrement Before LD**

first, decrement; then, store

POP

**Empty Ascending**




```
LDMDB r8! {r0,r1,r4}
LDMEA r8! {r0,r1,r4}
```

### LDM

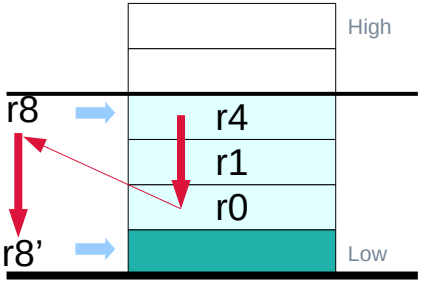
! auto index

**Decrement After LD**

first, store; then, decrement

POP

**Full Ascending**




```
LDMDA r8! {r0,r1,r4}
LDMFA r8! {r0,r1,r4}
```

## After

## References

[1] ftp://ftp.geoinfo.tuwien.ac.at/navratil/HaskellTutorial.pdf
[2] https://www.umiacs.umd.edu/~hal/docs/daume02yaht.pdf