MOV – Opcode

E,M -Operand

5E -HEX code

1 -Bytes

Data Transfer Group

Move(with byte)

MOV A, A 7F 1
MOV A, B 78 1
MOV A, C 79 1
MOV A, D 7A 1
MOV A, E 7B 1
MOV A, H 7C 1
MOV A, L 7D 1
MOVA, M 7E 1
MOV B, A 47 1
MOV B, B 40 1
MOV B, C 41 1
MOV B, D 42 1
MOV B, E 43 1
MOV B, H 44 1
MOV B,L 45 1
MOV B, M 46 1
MOV C, A 4F 1
MOV C, B 48 1
MOV C, C 49 1
MOV C, D 4A 1
MOV C, E 4B 1
MOV C, H 4C 1
MOV C, L 4D 1
MOV C, M 4E 1
MOV D, A 57 1
MOV D, B 50 1
MOV D, C 51 1
MOV D, D 52 1
MOV D, E 53 1
MOV D, H 54 1
MOV D, L 55 1
MOV D, M 56 1
MOV E, A 5F 1
MOV E, B 58 1
MOV E, C 59 1
MOV E, D 5A 1
MOV E, E 5B 1
MOV E, H 5C 1
MOV E, L 5D 1
MOV E, M 5E 1
MOVH,A 67 1
MOV H, B 60 1
MOV H, C 61 1
MOV H, D 62 1
MOV H, E 63 1
MOV H, H 64 1
MOV H, L 65 1
MOV H, M 66 1
MOV L, A 6F 1
MOV L, B 68 1
MOV L, C 69 1
MOV L, D 6A 1
MOV L, E 6B 1
MOV L, H 6C 1
MOV L, L 6D 1
MOV L, M 6E 1
MOV M, A 77 1
MOV M, B 70 1
MOV M, C 71 1
MOV M, D 72 1
MOV M, E 73 1
MOV M, H 74 1
MOV M, L 75 1

Move immediate(with byte)

MVI A, Data 3E 2
MVI B, Data 06 2
MVI C, Data 0E 2
MVI D, Data 16 2
MVI E, Data 1E 2
MVI H, Data 26 2
MVI L, Data 2E 2
MVI M, Data 36 2

LXI

Load Immediate
B,dble 01
D,dble 11
H, dble 21
SP,dble 31

Load/ Store *

LDAX B 0A
LDAX D 1A
LHLD adr2A
LDA adr3A
STAX B 02
STAX D 12
SHLD adr22
STA adr 32

XCHG

XCHG EB

byte = constact or logical/ arithmetic expression that evaluates to an 8-bit data quantity. (second byte of 2-byte instruction)

dble = constant or logical/ arithmetic expression that evaluates to a 16-bit data quantity.(second and third bytes of 3-bytes instructions).

adr= 16-bit address(second and third bytes of 3-byte instructions)
*= all flags (C, Z, S, P, AC) affected.
** = all flags except CARRY affected; (except INX and DCX affect no flags).
+ = only CARRY affected

Arithmetic and logical group

ADD* (with byte)

ADD A 87 1
ADD B 80 1 ADD C 81 1
ADD D 82 1
ADD E 83 1
ADD H 84 1
ADD L 85 1
ADD M 86 1

ADC(with byte)

ADC A 8F 1
ADC B 88 1
ADC C 89 1
ADC D 8A 1
ADC E 8B 1
ADC H 8C 1
ADC L 8D 1
ADC M 8E 1

Subtract*(with byte)

SUB A 97 1
SUB B 90 1
SUB C 91 1
SUB D 92 1
SUB E 93 1
SUB H 94 1
SUB L 95 1
SUB M 96 1

SBB(with byte)

SBB A 9F 1
SBB B 98 1
SBB C 99 1
SBB D 9A 1
SBB E 9B 1
SBB H 9C 1
SBB L 9D 1
SBB M 9E 1

DAD(double add) †

DAD B 09 1
DAD D 19 1 DAD H 29 1
DAD SP 39 1

Increment**(with Byte)

INR A 3C 1
INR B 04 1
INR C 0C 1
INR D 14 1
INR E 1C 1
INR H 24 1 INR L 2C 1
INR M 34 1

INX(with byte)

INX B 03 1 INX D 13 1
INX H 23 1
INX SP 33 1

DCR(with byte) **

DCR A 3D 1
DCR B 05 1
DCR C 0D 1
DCR D 15 1
DCR E 1D 1
DCR H 25 1
DCR L 2D 1
DCR M 35 1

DCX

DCX B 0B
DCX D 1B
DCX H 2B
DCX SP 3B

Specials

DAA* 27
CMA 2F
STC† 37
CMC† 3F

Arith and Logical Immediate

ADI byte C6
ACI byte CE
SUI byte D6
SBI byte DE
ANI byte E6
XRi byte EE
ORI byte F6
CPI byte FE

Rotate†

RLC 07
RRC 0F
RAL 17
RAR 1F

Logical* (With Bytes)

ANA A A7 1
ANA B A0 1
ANA C A1 1
ANA D A2 1
ANA E A3 1
ANA H A4 1
ANA L A5 1
ANA M A6 1

XRA(with byte)

XRA A AF 1
XRA B A8 1
XRA C A9 1
XRA D AA 1
XRA E AB 1
XRA H AC 1
XRA L AD 1
XRA M AE 1

ORA(with byte)

ORA A B7 1
ORA B B0 1
ORA C B1 1
ORA D B2 1
ORA E B3 1
ORA H B4 1
ORA L B5 1
ORA M B6 1

CMP(with byte)

CMP A BF 1
CMP B B8 1
CMP C B9 1
CMP D BA 1
CMP E BB 1
CMP H BC 1
CMP L BD 1
CMP M BE 1

BRANCH CONTROL GROUP

JUMP	
JMP adr	C3
JNZ adr JZ adr	C2
JNC adr	CA
JC adr	D2
JPO adr	DA
JPE adr	E2
JP adr	EA
JM adr	F2
PCHL	FA
	E9

CALL	
CALL adr	CD
CNZ adr	C4
CZ adr	CC
CNC adr	D4
CC adr	DC
CPO adr	E4
CPE adr	EC
CP adr	F4
CM adr	FC

Return	
RET	C9
RNZ RZ	C0
RNC	C8
RC	D0
RPO	D8
RPE	E0
RP	E8
RM	F0
	F8

Restart	
RST0	C7
RST1	CF
RST2	D7
RST3	DF
RST4	E7
RST5	EF
RST6	F7
RST7	FF

I/O AND MACHINE CONTROL

STACK Ops	
PUSH	
PUSH B	C5
PUSH D	D5 E5
PUSH H	F5
PUSH PSW	

POP	
POP B	C1
POP D	D1
POP H	E1
POP PSW*	F1
XTHL SPHL	E3 F9

Input/Output	
OUT byte IN byte	D3 DB

Control	
DI	F3
EI	FB
NOP	00
HLT	76

New Instuctions (8085 only)	
RIM SIM	20
	30

RESTART TABLE

Name	Code	Restart Address
RST 0	C7	0000H
RST 1	CF	0008H
RST 2	D7	0010H
RST 3	DF	0018H
RST 4	E7	0020H
TRAP	Hardware* Function	0024H
RST 5	EF	0028H
RST 5 5 RST 6	Hardware* Function	002CH
RST 6 5 RST 7	F7	003BH
RST 7 5	Hardware* Function	003CH
	FF	
	Hardware* Function	

*NOTE The hardware functions refer to the on-chip interrupt feature of the 8085 only

ASSEMBLER REFERENCE

Pseudo Instructions

General:

ORG
END
EQU
SET
DS
DB
DW

MACROS:

MACRO
ENDM
LOCAL
REPT
IRP
IRPC
EXITM

Relocation:

ASEG	NAME
DSEF	STKLN
CSEF	STACK
PUBLIC	MEMORY
EXTRN	

Conditional Assembly:

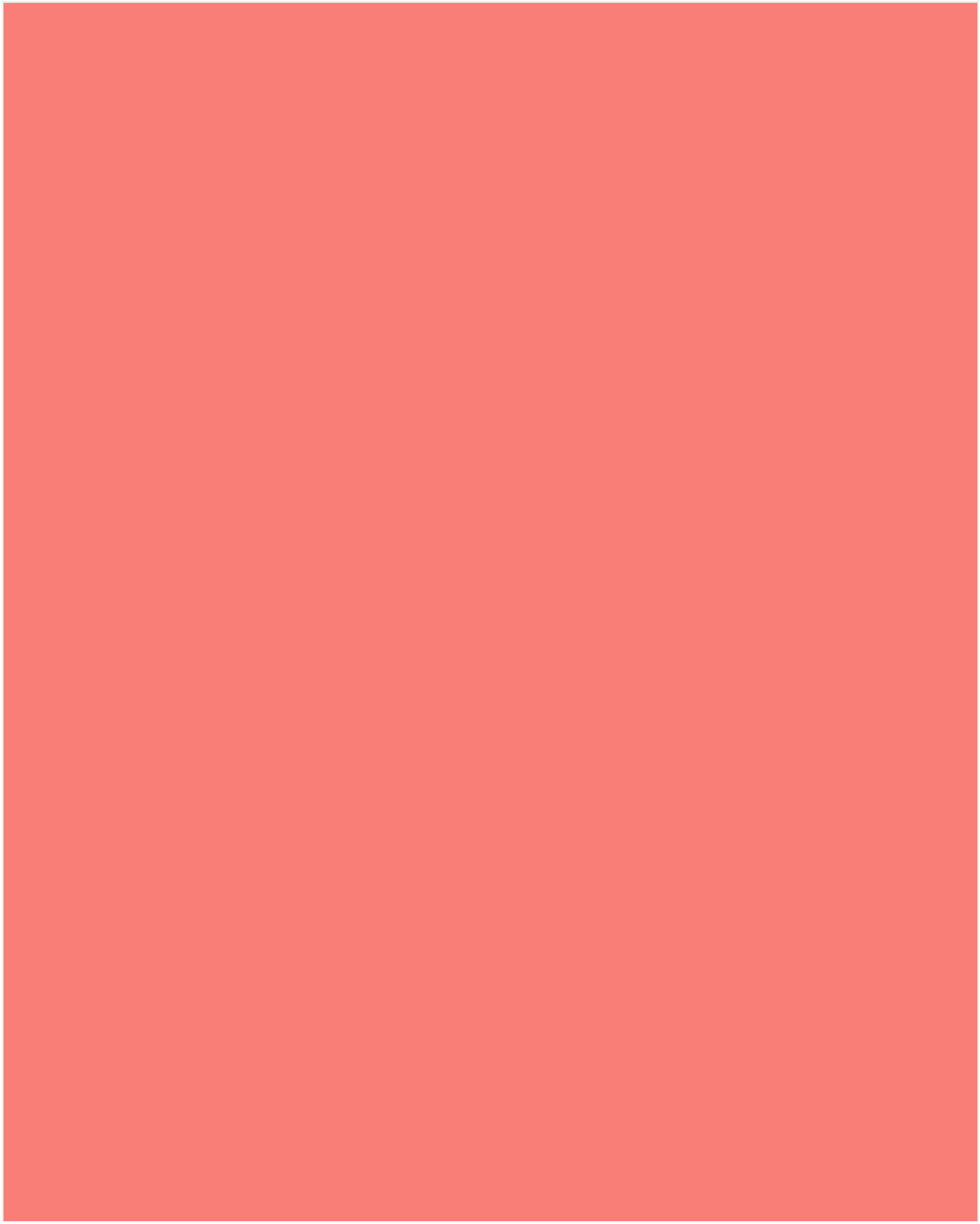
IF
ELSE
ENDIF

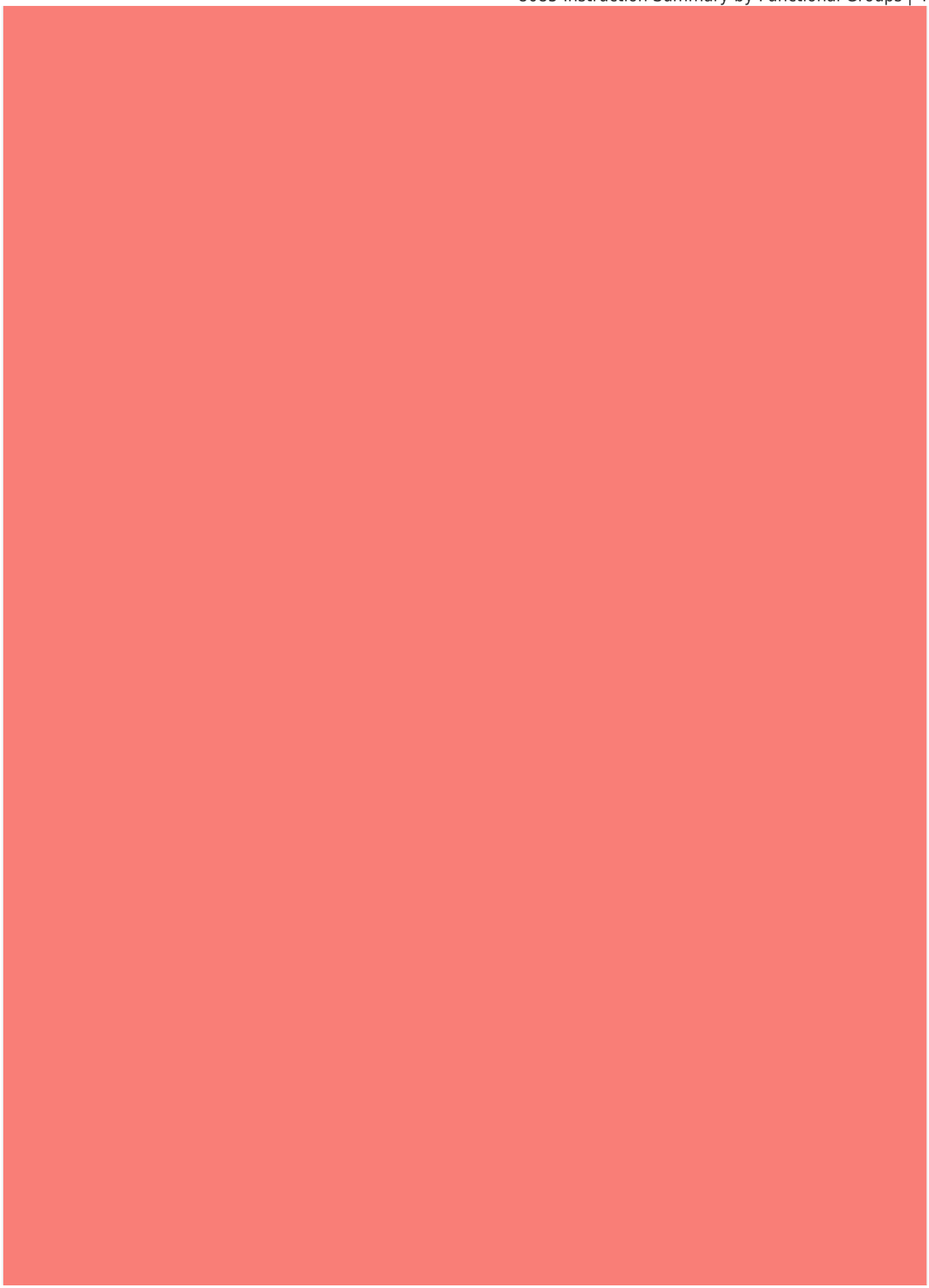
Manas Sharma

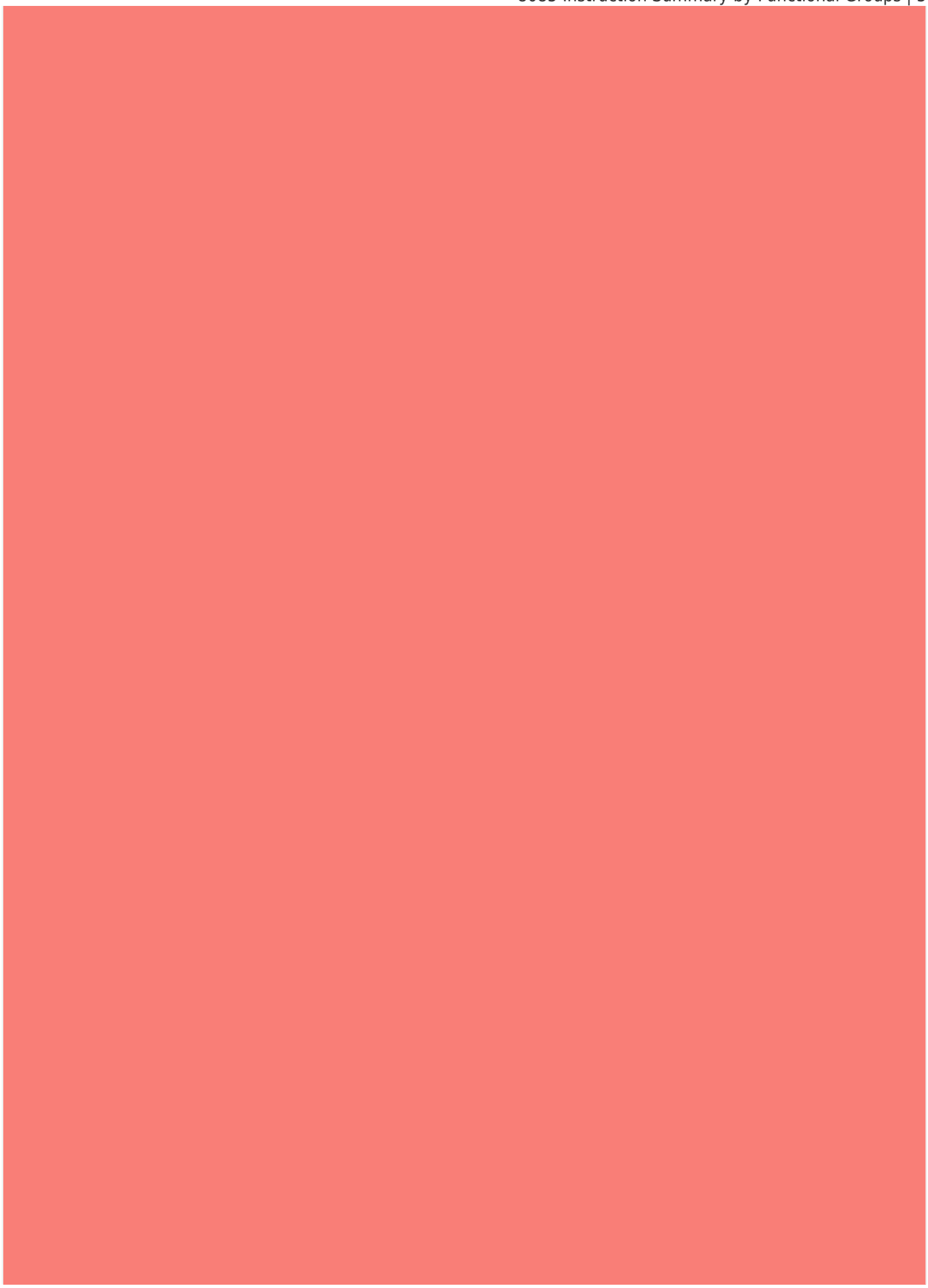
I’m a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time

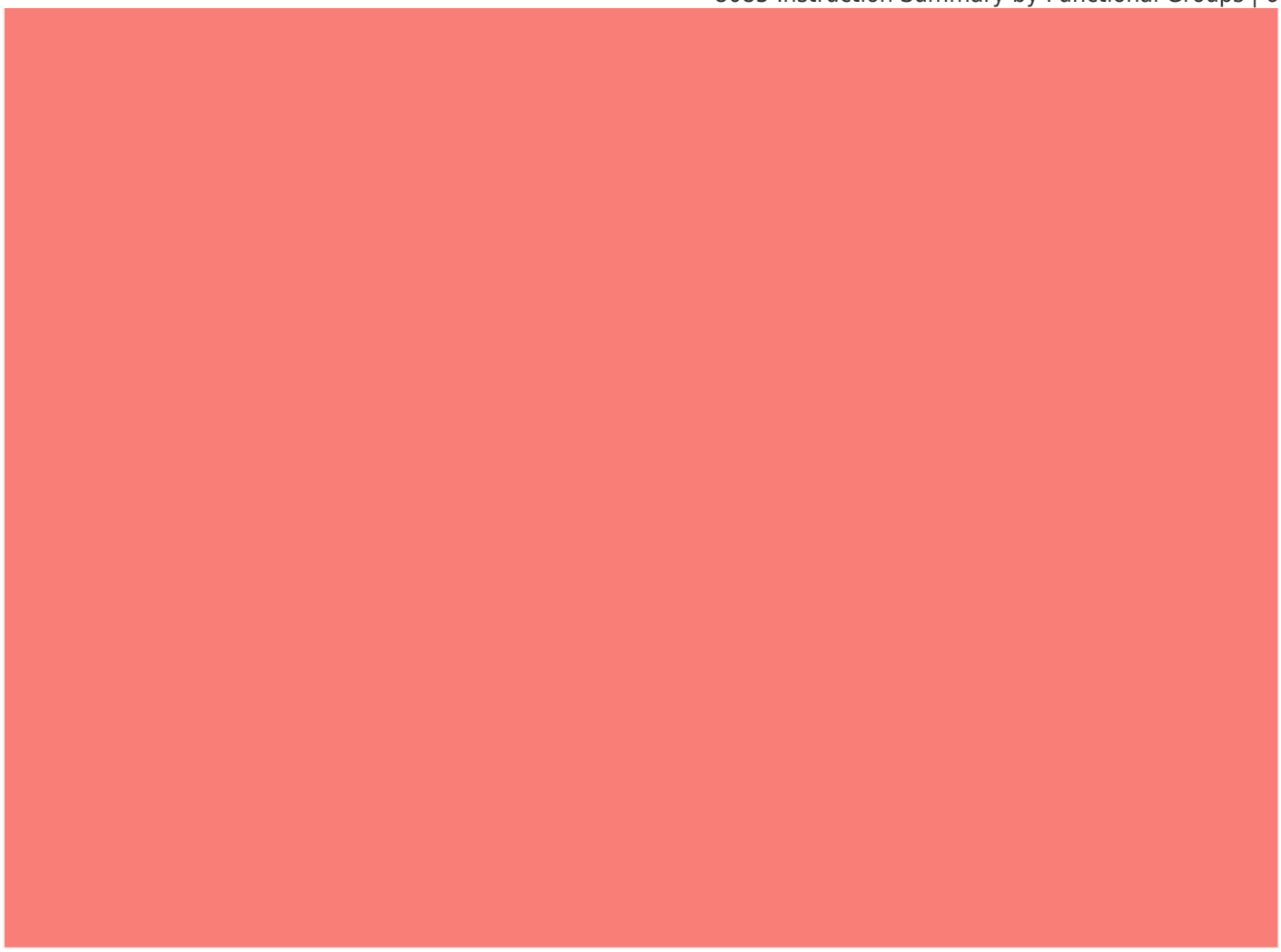
to time. Can code in most of the popular languages. I like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

manas.bragitoff.com/









Share this:

Click to share on Facebook (Opens in new window)

Click to share on Twitter (Opens in new window)

Click to share on WhatsApp (Opens in new window)

Click to share on Pinterest (Opens in new window)

Click to share on Reddit (Opens in new window)

Click to share on LinkedIn (Opens in new window)

Click to email a link to a friend (Opens in new window)

Click to print (Opens in new window)

Click to share on Tumblr (Opens in new window)

Click to share on Pocket (Opens in new window)

Click to share on Telegram (Opens in new window)

[wpedon id="7041" align="center"]