

INDICE

Unit 1. Introduction to Assembler Language	
Chapter 1. Meet the Computer	3
1.1. What is a Computer?	4
1.2. A Computer System	5
1.2.1. Storage Unit	6
1.2.2. The Arithmetic and Logical Unit (ALU)	
1.2.3. The Control Unit	12
1.2.4. Input	13
1.2.5. Output	16
1.2.6. Auxiliary Storage	16
1.3. Data Organization: Field, Record, File	19
1.4. Communicating with the Computer	21
1.4.1. Computer “Languages”	21
1.4.2. Machine Language	21
A. Registers	21
B. Addressing	23
C. Relative Addressing	24
D. The Machine Language Instruction Formats	28
1.4.3. Assembler Language	40
1.4.4. Why Assembler Language?	41
Chapter 2. Our First Program	42
2.1. Our First Problem – Described	50
2.2. The Program Instructions	56
2.2.1. Assembly Language Instructions: Assembler Instructions, Machine Instructions, and Macros	57
2.2.2. Instructions – Common Features	59
2.2.3. The Coding Form	62
2.3. Making the Computer Tick – Translating and Executing a Program	65
2.4. Explaining Mr. Sharp’s Program Instructions	66
2.5. Meet JCL – Job Control Language	96
2.6. The Program in its Entirety	98
2.6.1. On Coding Sheets	98
2.6.2. The Program Keypunched	101
2.6.3. Output of Assembler, Linkage Editor, and Program Execution	102
2.6.4. Recap: What Has the Program Done	108
Chapter 3. Defining and Defining Constants	113
3.1. How and Why is Data Placed into Storage?	114
3.2. The DC Instruction: Constant Definition	115
3.2.1. EBCDIC Constants	116
3.2.2. Packed Constants	122
3.2.3. Zoned Constants	127
3.2.4. Hexadecimal Constants	129
3.2.5. Binary Constants	131
3.3. Literals	135
3.4. The DS Instruction: Storage Definition	137
3.5. Our Second Program	141
3.6. The Road to Problem Solution	142

3.6.1. Understanding the Problem	143
Defining the Input	143
Defining the Output	145
Flowcharting	145
3.6.2. Coding the solution	150
Unit 2. The Decimal Instruction Set	
Chapter 4. Decimal Arithmetic – Operations With Packed Integer Values	165
4.1. A Payroll Problem	166
4.1.1. The Problem Described	166
4.1.2. Zoned vs Packed Forms	168
4.1.3 Payroll Program Logic	169
4.1.4. The Payroll Program – Coded	173
4.1.5. The Payroll Program – Output	179
4.1.6. What Do the Instructions Accomplish?	182
4.2. Explaining Our Program Instruction	186
4.2.1. The AP Instruction	186
4.2. 2. The SP Instruction	188
4.2.3. The ZAP Instruction	190
4.2.4. The MP Instruction	195
4.2.5. The DP Instruction	195
4.2.6. The PACK Instruction	200
4.2.7. The UNPK Instruction	203
Chapter 5. Simple Comparison and Simple Editing	211
5.1. Interpreting Program Output	212
5.2. Improving the Appearance Output	214
5.2.1. The MVZ Instruction	216
5.3. Comparison and Transfer of Control	219
5.3.1. Comparing Packed Numbers: The CP Instruction	220
5.3.2. Branching with Extended Mnemonics	224
5.3.3. The Payroll Program – Testing for Overtime	226
5.3.4. The Logical Compare in Main Storage: The CLC and CLI Instructions	233
Chapter 6. Decimal Arithmetic with Non – integer Values	243
6.1. Payroll Problem – Version 3	224
6.2. Techniques for Non – Integer Decimal Arithmetic	246
6.2.1. Adding and Subtracting Non – Integer Decimal Fields	246
6.2.2. Multiplying Non – Integer Decimal Fields	255
Truncation and Rounding	
6.2.3. Dividing Non – Integer Decimal Fields	260
6.3. Solving the Payroll Problem with Non – Integer Values	262
6.4. Instructions for Shifting Decimal Fields – The MVN, MVO, NC and NI Instructions	271
6.5. Shifting and Rounding Decimal Fields with SYSTEM /370 – The SRP Instruction	278
Unit 3. Binary and Hexadecimal Numbers, and the Standard Instruction Set	
Chapter 7. Working With Binary Numbers	291
7.1. The Standard Instruction	292

7.2. Positional Representation: Binary and Hexadecimal Numbers	293
7.2.1. The Binary Number System	294
7.2.2. The Hexadecimal Number System	309
7.2.3. Integer Conversions: Binary, Hexadecimal, and Decimal Numbers	314
7.3. Binary Numbers Within the Program	322
7.3.1. The Need for Conversion	322
7.3.2. Binary Numbers: Where?	323
7.3.3. Fixed * Point Constants, Storage Definitions, and Literals	325
7.3.4. A Program Example Using Binary Values	329
7.3.5. Binary Conversion Instructions: CVB, CVD	330
Chapter 8. Binary Integer Arithmetic	339
8.1. An Inventory Report Problem	340
8.1.1. The Problem Described	340
8.1.2. Inventory Report Program Logic	342
8.1.3. The Inventory Report Program – Coded	346
8.1.4. The Inventory Report Program – Output	346
8.1.5. How the instructions Work to Produce the Report	346
8.2. Explaining Our Program Instruction: Fixed – Point Arithmetic	353
8.2.1. Fixed – Point Addition Instructions: AR, A, AH	353
8.2.2. Fixed – Point subtraction Instruction: SR, S, SH	359
8.2.3. Fixed – Point Multiplication Instructions: MR, M, MH	364
8.2.4. Fixed – Point Division Instructions: DR, D	370
8.2.5. Preparing for Fixed – Point Division	373
8.2.6. Some Program Examples	376
8.2.7. Data Movement: Register to Register, Main Storage to Register, Register to Main Storage	378
Chapter 9. Binary Comparison, and Arithmetic with Nom – Integer Values	391
9.1. Inventory Report Problem – Version 2	391
9.2. Techniques for Non – Integer Fixed – Point Arithmetic	394
9.2.1. Adding and subtracting Non – Integer Fixed – Point Fields	394
9.2.2. Multiplying and Dividing Non – Integer Fixed – Point Fields	397
9.2.3. Using the Scale Factor in Binary Arithmetic	402
9.3. Comparing Fixed – Point Numbers	404
9.4. Solving the Inventory Report Problem with Non – Integer Values	408
9.5. Algebraic Shift Instructions	419
Unit 4. Editing, and More Advanced Branching and Looping	
Chapter 10. Toward More Readable Printed Output: Editing and Forms Control	433
10.1. The Edit Instruction	434
10.1.1. Suppressing Leading Zeros	435
10.1.2. Punctuating Numeric Fields	439
10.1.3. The significance Starter: Establishing Early significance	441
10.1.4. Signed Fields: A Negative Suffix	443
10.1.5. Editing Several Fields with One Instruction	445
10.1.6. The ED Instruction Summarized	449
10.2. The Edit and Mark Instruction	450
10.2.1. Fixed vs Floating Symbols	450
10.2.2. Using the EDMK instruction	452

10.2.3. The EDMK instruction Summarized	456
10.3. Forma Control and Stoker Selection	458
10.3.1. Carriage Control	459
10.3.2. The CNTRL Macro Instruction	460
10.3.3. The PRTOV macro Instruction	464
10.3.4. Carriage Control on spooled systems	466
10.4. The COMRG and TIME macro Instructions	467
10.5. A report Program with Editing and forms Control	469
Chapter 11. More On Branching and Looping	481
11.1. A Social Security Problem	482
11.1.1. The Problem Described	482
11.1.2. The Condition Code and Instruction Address Register	485
11.1.3. The BC and BCR Instructions	486
11.1.4. The Social Security Problem – Coded	489
11.1.5. The Extended Mnemonic Instructions	493
11.2. A Table Handling Problem	494
11.2.1. What Are Tables?	495
11.2.2. The Problem Described	496
11.2.3. Loop Structure	497
11.2.4. Using Base Registers and Index Registers for table Reference	499
11.2.5. The Table Problem with Branch on Index Instructions: BXH an BXLE	507
11.2.6. Loop Control with the BCT and BCTR Instructions	514
Unit 5. Operating System Concepts	
Chapter 12. Subroutines and Subroutine Linkage	525
12.1. The Need for Subroutines	526
12.1.1. A Programming Dilemma: Providing Output Headings page by page	526
12.1.2. Internal Subroutines	531
12.2.1. Entry To and Return From Internal Subroutines: The BAL, BALR, and BR Instructions	531
12.2.2. Our Heading Program Coded: An Example Using Subroutines	535
12.3. External Subroutines	537
12.3.1. The Need for Linkage Conventions	537
12.3.2. Entry to and Return from external subroutines	538
12.3.3. Passage of Data Between Calling and Called Routines	543
12.3.4. Saving and Restoring Register Contents	547
12.3.5. Summary: Responsibilities of Calling and of Called Routine	552
12.3.6. The SAVE, RETURN and CALL Macros	556
12.3.7. A Program Example	559
12.4. The Linkage Editor and Subroutine Linkage	559
Chapter 13. Some Operating System Features: Virtual Storage, the Program Status Word, the Interrupt system	567
13.1. What is an Operating System?	568
13.2. Virtual Storage Systems	569
13.2. Storage allocation	570
Relocation	570
Static relocation	572
Dynamic relocation	573

13.2. Techniques for dynamic relocation: Segmentation and Paging	
A segmentation system	573
A segmentation and paging system	577
13.2.3. Virtual storage (VS): Its meaning and Its structure	582
13.2.4. Program Execution in a virtual storage system	584
13.2.5. Single virtual storage vs multiple virtual storages	588
13.2.6. Coding for virtual storage	589
13.2.7. Advantages of virtual storage	590
13.3. Some Key Features of System Design	591
13.3.1. The Program Status Word in BC Mode	591
13.3.2. The program Status Word in EC Mode	595
13.3.3. The Interrupt System	598
13.4. The IBM 3033 Processor	606
Chapter 14. Debugging	615
14.1. The Best Debugging Medicine is Preventive	616
14.2. Desk – Checking	617
14.3. Diagnostics	618
14.4. Testing the Program	627
14.4.1. Test Data	632
14.4.2. Program Interruptions: The Program – Check message	633
14.4.3. Program Dumps	637
14.4.4. Analyzing a Dump: A Debugging Example	656