

Table of Contents

Preface	xiii
PART 1. ELEMENTS OF A BASIC ARCHITECTURE	1
Chapter 1. Introduction	3
1.1. Historical background	3
1.1.1. Automations and mechanical calculators	3
1.1.2. From external program to stored program	6
1.1.3. The different generations	8
1.2. Introduction to internal operation	13
1.2.1. Communicating with the machine	13
1.2.2. Carrying out the instructions	14
1.3. Future prospects	15
Chapter 2. The Basic Modules	17
2.1. Memory	17
2.1.1. Definitions	17
2.1.2. A few elements of technology	19
2.2. The processor	20
2.2.1. Functional units	20
2.2.2. Processor registers	21
2.2.3. The elements of the processing unit	26
2.2.4. The elements of the control unit	28
2.2.5. The address calculation unit	29
2.3. Communication between modules	30
2.3.1. The PCI bus	31
Chapter 3. The Representation of Information	35
3.1. Review	36

3.1.1. Base 2	36
3.1.2. Binary, octal and hexadecimal representations	37
3.2. Number representation conventions	38
3.2.1. Integers	38
3.2.2. Real numbers	40
3.2.3. An example of a floating-point representation, the IEEE-754 standard	44
3.2.4. Dynamic range and precision	46
3.2.5. Implementation	47
3.2.6. Extensions of the IEEE-754 standard	47
3.3. Character representation	48
3.3.1. 8-bit representation	48
3.3.2. Modern representations	50
3.4. Exercises	52
PART 2. PROGRAMMING MODEL AND OPERATION	55
Chapter 4. Instructions	57
4.1. Programming model	58
4.1.1. The registers of the I8086	58
4.1.2. Address construction and addressing modes	59
4.2. The set of instructions	62
4.2.1. Movement instructions	62
4.2.2. Arithmetic and logic instructions	62
4.2.3. Shift instructions	64
4.2.4. Branching	65
4.2.5. Other instructions	67
4.3. Programming examples	68
4.4. From assembly language to basic instructions	70
4.4.1. The assembler	70
4.4.2. The assembly phases	72
4.4.3. The linker	73
4.4.4. When to program in assembly language	74
Chapter 5. The Processor	75
5.1. The control bus	76
5.1.1. Reset line	77
5.1.2. Hold line	77
5.1.3. Wait control line	78
5.1.4. Interrupt lines	78
5.1.5. Conceptual diagram	78
5.2. Execution of an instruction: an example	79
5.2.1. Execution of the instruction	81

5.2.2. Timing diagram	85
5.3. Sequencer composition	87
5.3.1. Traditional synthesis methods	87
5.3.2. Microprogramming	90
5.4. Extensions	91
5.4.1. Coprocessors	91
5.4.2. Vector extensions	94
5.4.3. DSP and GPU	99
5.5. Exercise	101
Chapter 6. Inputs and Outputs	103
6.1. Examples	105
6.1.1. Example: controlling a thermocouple	105
6.1.2. Example: serial terminal connection	111
6.2. Design and addressing of EU	115
6.2.1. Design of exchange units	115
6.2.2. Exchange unit addressing	116
6.3. Exchange modes	118
6.3.1. The polling exchange mode	118
6.3.2. Direct memory access	119
6.3.3. Interrupts	126
6.4. Handling interrupts	127
6.4.1. Operating principle	127
6.4.2. Examples	129
6.4.3. Software interrupts	131
6.4.4. Masking and unmasking interrupts	131
6.4.5. Interrupt priorities or levels	132
6.4.6. Similar mechanisms	132
6.5. Exercises	133
PART 3. MEMORY HIERARCHY	137
Chapter 7. Memory	139
7.1. The memory resource	139
7.2. Characteristics	140
7.3. Memory hierarchy	141
7.3.1. Principle of locality	142
7.3.2. Hierarchy organization and management	143
7.3.3. Definitions and performance	144
7.4. Memory size and protection	145
7.5. Segmentation	145
7.5.1. Using segment registers: an example	146
7.5.2. Using segment descriptors	147

7.6. Paging	148
7.7. Memory interleaving and burst mode	151
7.7.1. C-access	151
7.7.2. S-access	151
7.7.3. Burst mode	153
7.7.4. Prefetch buffers	153
7.8. Protections, example of the I386	154
Chapter 8. Caches	157
8.1. Cache memory	157
8.1.1. Operating principle and architectures	157
8.1.2. Cache memory operation	158
8.1.3. Cache design	160
8.2. Replacement algorithms	165
8.2.1. The LRU method	165
8.2.2. The case of several levels of cache	171
8.2.3. Performance and simulation	172
Chapter 9. Virtual Memory	175
9.1. General concept	176
9.1.1. Operation	176
9.1.2. Accessing information	176
9.1.3. Address translation	177
9.2. Rules of the access method	178
9.2.1. Page fault	178
9.2.2. Multi level paging	179
9.2.3. Service information, protection and access rights	180
9.2.4. Page size	181
9.3. Example of the execution of a program	182
9.3.1. Introducing the translation cache	184
9.3.2. Execution	184
9.3.3. Remarks	187
9.4. Example of two-level paging	188
9.4.1. Management	188
9.4.2. Handling service bits	190
9.4.3. Steps in the access to information	190
9.5. Paged segmentation	194
9.5.1. 36-bit extensions	196
9.6. Exercise	197
9.7. Documentation excerpts	198
9.7.1. Introduction to the MMU	198
9.7.2. Description of the TLB	199
9.7.3. TLB features	202

PART 4. PARALLELISM AND PERFORMANCE ENHANCEMENT	205
Chapter 10. Pipeline Architectures	207
10.1. Motivations and ideas	207
10.1.1. RISC machines	207
10.1.2. Principle of operation	209
10.1.3. Cost of the pipeline architecture	211
10.2. Pipeline management problems	212
10.2.1. Structural hazards	212
10.2.2. Dependency conflicts	216
10.2.3. Branches	217
10.3. Handling branches	218
10.3.1. Delayed branches and software handling	218
10.3.2. Branch predictions	221
10.3.3. Branch target buffer	225
10.3.4. Global prediction	227
10.3.5. Examples	229
10.4. Interrupts and exceptions	233
10.4.1. Interrupts	234
10.4.2. Traps and faults	234
Chapter 11. Example of an Architecture	235
11.1. Presentation	235
11.1.1. Description of the pipeline	235
11.1.2. The instruction set	238
11.1.3. Instruction format	239
11.2. Executing an instruction	240
11.2.1. Reading and decoding an instruction	240
11.2.2. Memory read	241
11.2.3. Memory write operations	241
11.2.4. Register to register operations	242
11.2.5. Conditional branching	243
11.2.6. Instruction with immediate addressing	246
11.3. Conflict resolution in the DLX	246
11.3.1. Forwarding techniques	247
11.3.2. Handling branches	249
11.4. Exercises	252
Chapter 12. Caches in a Multiprocessor Environment	261
12.1. Cache coherence	262
12.1.1. Examples	262
12.1.2. The elements to consider	264

12.1.3. Definition of coherence	264
12.1.4. Methods	265
12.2. Examples of snooping protocols	267
12.2.1. The MSI protocol	267
12.2.2. The MEI protocol	270
12.2.3. The MESI protocol	271
12.2.4. The MOESI protocol	273
12.3. Improvements	275
12.4. Directory-based coherence protocols	275
12.5. Consistency	278
12.5.1. Consistency and coherence	278
12.5.2. Notations	279
12.5.3. Atomic consistency	280
12.5.4. Sequential consistency	281
12.5.5. Causal consistency	282
12.5.6. Weak consistency	283
12.6. Exercises	284
Chapter 13. Superscalar Architectures	287
13.1. Superscalar architecture principles	287
13.1.1. Hazards	288
13.2. Seeking solutions	290
13.2.1. Principles	290
13.2.2. Example	293
13.3. Handling the flow of instructions	295
13.3.1. Principle of scoreboarding	295
13.3.2. Scoreboarding implementation	296
13.3.3. Detailed example	297
13.3.4. Comments on precedence constraints	302
13.3.5. Principle of the Tomasulo algorithm	303
13.3.6. Detailed example	306
13.3.7. Loop execution and WAW hazards	313
13.4. VLIW architectures	315
13.4.1. Limits of superscalar architectures	315
13.4.2. VLIW architectures	316
13.4.3. Predication	317
13.5. Exercises	321
PART 5. APPENDICES	325
Appendix A. Hints and Solutions	327
A1.1. The representation of information	327
A1.2. The processor	330

A1.3. Inputs and outputs	331
A1.4. Virtual memory	333
A1.5. Pipeline architectures	335
A1.6. Caches in a multiprocessor environment	341
A1.7. Superscalar architectures	344
Appendix B. Programming Models	347
A2.1. Instruction coding in the I8086	347
A2.2. Instruction set of the DLX architecture	349
A2.2.1. Operations on floating-point numbers	349
A2.2.2. Move operations	349
A2.2.3. Arithmetic and logic operations	350
A2.2.4. Branches	350
Bibliography	351
Index	357