ISA 2

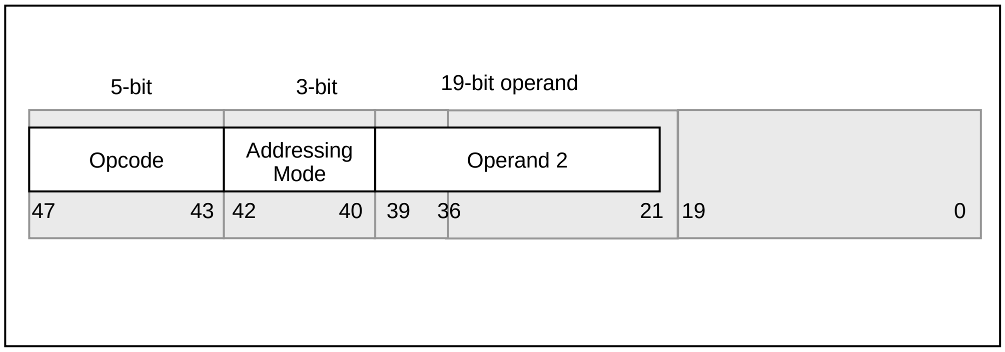
ACCUMULATOR ARCHITECTURE

Abbreviations

Accumulator	→ Acc
Carry Flag	→ CF
Over-flow flag	→ OF
Sign Flag	→ SF
Zero Flag	→ ZF
Parity Flag	→ PF
Instruction Pointer	→ IP

Instruction format

General Instruction format



Operand addressing modes and notations

000	→	Immediate	#
100	→	Direct	default
101	→	Indirect	&

Width of :

a memory location is 16 bits

an instruction is 48 bits
an instruction = width three memory locations

the processor data bus is 16 bits
the Processor address bus is 19 bits

the accumulator is 16 bits.

Implied addresses

Return_address (used in Ret instruction)	20000H
Loop counter (used in Looz instruction)	20001H

Instruction Set Summary

Instruction set reference

Arithmetic instructions

ADD	Addition
Opcode	10000
Format	ADD AM operand
Description	Adds the operand specified and the Acc contents. Result modifies the Acc. The ADD instruction performs integer addition. It evaluates the results for both signed and unsigned integer operands and sets the CF and OF flags to indicate a carry in the signed or unsigned result, respectively. The SF indicates the sign of the signed result.
Operation	Acc ← Acc + Operand
AMs	Immediate, direct and indirect
Flags affected	ZF, OF, SF and PF are set according to the result.

SUB	Subtraction
Opcode	10001
Format	SUB AM operand
Description	Subtracts the Acc contents from the operand specified. Result modifies the Acc. The SUB instruction performs integer subtraction. It evaluates the results for both signed and unsigned integer operands and sets the CF and OF to indicate a carry in the signed or unsigned result, respectively. The SF indicates the sign of the signed result.
Operation	Acc ← Acc – Operand
AMs	Immediate, direct and indirect
Flags affected	ZF, OF, SF, CF and PF are set according to the result.

MUL Signed multiplication

Opcode	11011
Format	MUL AM operand (lower 8-bit)
Description	Performs signed multiplication on lower 8-bits of accumulator and the operand. Mul instruction produces a 16-bit result in the accumulator.
Operation	Acc ← Acc * Operand (16-bit) (8 LSBs) (8 LSBs)
AMs	Immediate, direct and indirect
Flags affected	ZF, SF and PF are set according the product. OF and CF is cleared, as no overflow occurs in this situation and no carry considered.

DIV Signed division

Opcode	11100
Format	DIV AM operand
Description	Divides the accumulator by the operand. Div instruction produces an integer output. Result modifies the accumulator.
Operation	Acc ← Acc / operand
AMs	Immediate, direct and indirect
Flags affected	ZF, SF and PF are set according the product. OF and CF is cleared, as no overflow occurs in this situation and no carry considered.

INC	Increment Acc by 1
Opcode	10101
Format	INC
Description	Adds 1 to the Acc. Evaluates for both signed and unsigned operands.
Operation	Acc ← Acc +1
AMs	Immediate, direct and indirect
Flags affected	the CF is not affected. The OF, SF, ZF and PF are set according to the result.

Logical instructions

AND	Bit-wise And
Opcode	10010
Format	AND AM operand
Description	Performs bit wise AND operation on the specified operand and the Acc. Each bit of the result is set to 1 if both corresponding bits of the operands are 1; otherwise, it is set to 0. Result replaces the Acc.
Operation	Acc ← Acc AND specified operand
AMs	Immediate, direct and indirect
Flags affected	OF and CF are cleared; SF, ZF and PF are set according to the result.

OR Bit-wise OR

Opcode	10011
Format	OR AM operand
Description	Performs bit wise OR operation on the specified operand and the Acc. Each bit of the result is set to 0 if both corresponding bits of the operands are 0; otherwise, it is set to 1. Result replaces the Acc.
Operation	Acc ← Acc OR specified operand
AMs	Immediate, direct and indirect
Flags affected	OF and CF are cleared; SF, ZF and PF are set according to the result.

XOR Bit-wise XOR

Opcode	10100
Format	XOR AM operand
Description	Performs bit wise XOR operation on the specified operand and the Acc. Each bit of the result is set to 0 if both corresponding bits of the operands are 1 or if both corresponding bits of the operands are 0; otherwise, it is set to 1. result replaces the Acc.
Operation	Acc ← Acc XOR specified operand
AMs	Immediate, direct and indirect
Flags affected	OF and CF are cleared; SF, ZF and PF are set according to the result.

SHL Shift Acc left 1-bit

Opcode	10111
Format	SHL
Description	Shifts the bits in the Acc to the left by one bit. Bits beyond the boundary are first shifted in to the CF. At the end of the shift operation CF contains the MSB of the Acc. Least significant bit is cleared.

Operation	Acc ← Acc (MSB-1 down to 0) & "0"
	CF ← MSB (original operand)

AMs	Implied (accumulator)
Flags affected	The CF contains the value of the bit shifted out of the Acc. OF is set to zero if the value of the MSB of the Acc is same as the CF; otherwise it is set to 1. SF, ZF and PF are set according to the result.

SHR Shift Acc right 1-bit

Opcode	01010
Format	SHR
Description	Shifts the bits in the Acc to the right by one bit. Bits beyond the boundary are first shifted in to the CF. At the end of the shift operation CF contains the LSB of the Acc. Most significant bit is cleared.

Operation	Acc ← "0" & Acc (MSB down to 1)
	CF ← LSB (Acc)

AMs	Implied (accumulator)
Flags affected	The CF contains the value of the bit shifted out of the Acc. OF is set to the MSB of the original operand. SF, ZF and PF are set according to the result.

ROL rotate Acc left 1-bit

Opcode	11001
Format	ROL
Description	shifts (rotates) the bits of the Acc left by 1 bit. Instruction includes the CF in the rotation, first shifts the CF into the LSB. And shifts the MSB to the CF. The original value of the CF are not a part of the result, but the CF receives a copy of the bit that was shifted from one end to the other end. OF is set to the exclusive OR of the CF (after the rotate) and the MSB of the result.

Operation	tempCF ← MSB (operand)
	Acc ← Acc (msb-1 down to 0) & CF
	CF ← tempCF
	OF ← MSB (operand) XOR CF (after the rotation)

AMs	Implied (accumulator)
Flags affected	The CF contains the value of the bit shifted out of Acc. OF is set to the exclusive OR of the CF (after the rotate) and the MSB of the result. SF, ZF and PF are set according to the result.

ROR rotate Acc right 1-bit

Opcode	11010
Format	ROR
Description	shifts (rotates) the bits of the Acc right by 1 bit. Instruction includes the CF in the rotation, first shifts the CF into the MSB and shifts the LSB to the CF. The original value of the CF is not a part of the result, but the CF receives a copy of the bit that was shifted from one end to the other end. OF is set to the exclusive OR of the two most significant bits of the Acc.

Operation	tempCF ← LSB (Acc)
	MSB (Acc) ← CF
	CF ← tempCF
	OF ← MSB (Acc) XOR CF

AMs	Implied (accumulator)
Flags affected	The CF contains the value of the bit shifted into it. OF is set to the exclusive-OR of the two most significant bits of the Acc. SF, ZF and PF are set according to the result.

NOT One’s compliment negation

Opcode	10110
Format	NOT
Description	Performs bit-wise NOT operation (each 1 is set to 0, and each 0 is set to 1) on the Acc and stores the result in it.
Operation	Acc ← NOT Acc
AMs	Implied (accumulator)
Flags affected	none

Control Transfer

Conditional Branches

JC	Jump if carry (CF = 1)
Opcode	00001
Format	JC {Signed offset (IP relative)}
Description	Check the state of the CF in the PSW (Programmer's Status Word register) and, if the flag is set, performs a jump to the target instruction specified by the operand. If the condition is not satisfied, the jump is not performed and execution continues with the instruction following the JC instruction. The target instruction is specified with a relative offset (a signed offset relative to the current value of the IP register). Machine code level the offset is encoded as a signed immediate value, which is added to the IP.
Operation	IP $\leftarrow$ IP + Operand
AM	IP relative addressing is applied in the address calculation.
Flags affected	none

JOF	Jump if an over-flow (OF = 1)
Opcode	00010
Format	JOF {Signed offset (IP relative)}
Description	Check the state of the OF in the PSW (Programmer's Status Word register) and, if the flag is set, performs a jump to the target instruction specified by the operand. If the condition is not satisfied, the jump is not performed and execution continues with the instruction following the JOF instruction.
Operation	IP $\leftarrow$ IP + Operand
AM	IP relative addressing is applied in the address calculation.
Flags affected	none

JS Jump if Sign (SF = 1)

Opcode	00011
Format	JS {Signed offset (IP relative)}
Description	Check the state of the SF in the PSW (Programmer's Status Word register) and, if the flag is set, performs a jump to the target instruction specified by the operand. If the condition is not satisfied, the jump is not performed and execution continues with the instruction following the JS instruction
Operation	IP $\leftarrow$ IP + Operand
AM	IP relative addressing is applied in the address calculation.
Flags affected	none

JP Jump if parity (PF = 1)

Opcode	00100
Format	JP {Signed offset (IP relative)}
Description	Check the state of the PF in the PSW (Programmer's Status Word register) and, if the flag is set, performs a jump to the target instruction specified by the operand. If the condition is not satisfied, the jump is not performed and execution continues with the instruction following the JP instruction
Operation	IP $\leftarrow$ IP + Operand
AM	IP relative addressing is applied in the address calculation.
Flags affected	none

JZ	Jump if result is zero (ZF =0)
Opcode	00101
Format	JZ {Signed offset (IP relative)}
Description	Check the state of the ZF (Programmer's Status Word register) and, if set, performs a jump to the target instruction specified by the operand. If the condition is not satisfied, the jump is not performed and execution continues with the instruction following the JZ instruction
Operation	IP ← IP + Operand
AM	IP relative addressing is applied in the address calculation.
Flags affected	none

Unconditional branch

JUMP	Jump
Opcode	01111
Format	JUMP {Signed offset (IP relative)}
Description	Transfers the program control to a different point of the instruction stream. The operand specified the signed offset being jumped to. This operand is an immediate signed offset
Operation	IP ← IP + operand
AM	IP relative addressing is applied in the address calculation.
Flags affected	none

Loop Instructions

LOOZ	Loop while zero
Opcode	01000
Format	LOOZ {Signed offset (IP relative)}
Description	Performs a loop operation using an implied memory location as a counter. Each time the LOOZ instruction executes, the counter is decremented, then checked for 0. If the count =0, the loop terminates and the program execution continues with the instruction following the LOOZ instruction. If the count is not 0, a jump is performed to the specified operand, (a signed offset is specified in the instruction) which is presumably the instruction at the beginning of the loop. The decremented counter value is written back to the implied memory location whenever it is nor zero.
Operation	see below
AM	IP relative addressing is applied in the address calculation.
Flags affected	none

Operation:

```
Count ← Count – 1
IF Count =0
    Loop termination
ELSE
    IP ← IP + operand
END IF
```

Calls and Returns

CALL	Call procedure
Opcode	01010
Format	CALL {Immediate address}
Description	Saves IP in the implied memory location and branches to the procedure (called procedure) specified by the operand. This operand is an immediate value. When executing a CALL, the processor pushes the value of the IP register on to the implied return address (For use latter as a return-instruction pointer). The processor then branches to the address specified with instruction.
Operation	implied address ← IP IP ← IP + Operand
AM	Immediate operand is used as the jumping location
Flags affected	none

RETURN	Return from procedure
Opcode	01011
Format	RET
Description	Transfers program control to a return address located in the implied return address. Return is made to the instruction that follows the CALL instruction.
Operation	IP ← implied return address
AM	Contents of the implied return address is used as the jumping location
Flags affected	none

Miscellaneous instructions

NOP	No operation
Opcode	01110
Format	NOP
Description	Performs no operation. Takes up space in the instruction stream but does not affect the context, except IP.
Flags affected	none

Data Movement instructions

LOADacc Move to Acc

Opcode	11111
Format	LOADacc AM operand
Description	**Immediate Addressing** Copy the 16-bit immediate source operand to the accumulator. **Direct Addressing** Copy the contents of the source (operand) address (of memory) to the accumulator.

Operation	**Immediate Addressing**
	Acc ← 16-bit immediate Operand
	
	Direct Addressing
	Acc← memory (operand)

AMs	Immediate and direct
Flags affected	none

STOREacc Move from Acc

Opcode	00110
Format	STOREacc AM operand
Description	**Direct Addressing** Copy the accumulator to the destination memory address (operand). **Indirect Addressing** Copy the contents of the accumulator to the destination memory address found at operand (in memory).
Operation	**Direct Addressing** Memory (operand) ← Acc **Indirect Addressing** Memory {memory (operand)} ← Acc
AMs	Direct and indirect
Flags affected	none