

Motorola 6800 ir Intel 4040 architektūrų palyginimas

Parengė Ieva Lileikytė, 1 grupė, 2 pogrupis

https://github.com/IevaLil/Palyginimo_uzduotis

- **Kokia elementinė kompiuterio/procesoriaus bazė? Kokios fizinės įrangos savybės (svoris, dydis, energijos suvartojimas)?**

Motorola 6800 ir Intel 4040 yra mikroprocesoriai. Abu išleisti 1974 metais.

Motorola 6800 mikroprocesorius pagamintas naudojant “DIP 40-pin” (“Dual In-Line Package”) 40 kontaktų korpusą. Tokio korpuso dydis yra maždaug: 14mm × 52 mm × 5mm. Turi 4.100 tranzistorių.

Svoris: apie 23 gramus. Energijos suvartojimas: apie 1.3W.

Intel 4040 mikroprocesorius pagamintas naudojant “DIP 24-pin” korpusą. Tokio korpuso matmenys yra maždaug: 18mm × 31mm × 5mm. Turi 3.000 tranzistorių.

Svoris: apie 15 gramų. Energijos suvartojimas: mažesnis nei 0.5W.

- **Kokio tipo architektūra turėjo abu kompiuteriai?**

Ir Motorola 6800 procesorius ir Intel 4040 remiasi registrine architektūra.

“The architecture of the 4004 was a fairly straightforward register-based architecture” (Intel4004.com, 2024).

Intel 4040 yra patobulinta versija Intel 4004.

- **Ar tai buvo beadresinės (stekinės), vieno adreso, dviejų adresų, trijų adresų ar keturių adresų mašinos?**

Motorola 6800 yra dviejų adresų mašina, o Intel 4040 yra vieno adreso mašina.

- **Kokie registrai šiose architektūrose?**

Registrai Motorola 6800:

Akumuliatorius A (ACCA): 8 bitų registras, naudojamas aritmetinėms ir loginėms operacijoms.

Akumuliatorius B (ACCB): 8 bitų registras, naudojamas aritmetinėms ir loginėms operacijoms.

Indekso registras (IX): 16 bitų registras, dažniausiai naudojamas laikiniams duomenims saugoti arba kaip indeksas, kai naudojamas indeksuotas adresavimas.

Programos skaitiklis (PC): 16 bitų registras atsakingas už dabartinio vykdomo kodo adresavimą.

Steko rodyklė (SP): 16 bitų registras. Naudojamas funkcijų iškvietimams ir grįžimams stebėti.

Vėliavų registras (Condition Code Register arba CCR):

- Tarpinių bitų pernešimai (H): nustatoma, jei skaičiuojant rezultatą buvo perkelta iš 3 į 4 bitą.
- Pertraukimai (I): nustatoma, jei IRQ pertraukimas išjungtas.
- Neigiamas (N): nustatomas, jei nustatytas reikšmingiausias rezultato bitas.
- Nulis (Z): nustatomas, jei rezultatas lygus nuliui.
- Perpildymas (V): nustatomas, jei paskutinio rezultato skaičiavimo metu buvo perpildymas.
- Carry (C): nustatomas, jei paskutinio rezultato skaičiavimo metu buvo perkėlimas iš 7 bito.

Iš viso 6 pagrindiniai registrai.

Registrai Intel 4040:

Programos skaitiklis (PC): 12 bitų.

Steko registrai: “There are seven stack level registers, which is sufficient to implement 7-level deep subroutine calls. Every subroutine call stores return address in one of the stack registers. Contents of the accumulator and the index registers is not preserved. When the program is interrupted, the contents of the send register (SRC) is also preserved in the stack.” (CPU-World, 2024)

Akumulatorius (ACC): 4 bitų, aritmetinėms operacijoms, loginėms operacijoms, taip pat naudojamas keistis duomenimis su RAM ir I/O.

Indekso registrai: “The microprocessor has 24 4-bit registers, organized as 16 registers in bank 0 and 8 registers in bank 1. Lower 8 registers can be selected from bank 0 or 1 using SB0 and SB1 instructions. The registers can work in pairs as 8 8-bit registers.” (CPU-World, 2024)

Iš viso 33 registrai.

- **Ar požymių bitai buvo naudojami šiose architektūrose? Kokie požymiai buvo naudojami?**

Abi architektūros naudojo požymių bitus.

Motorola 6800

Vėliavų registras (Condition Code Register arba CCR):

- Tarpinių bitų pernešimai (H): nustatoma, jei skaičiuojant rezultatą buvo perkelta iš 3 į 4 bitą.
- Pertraukimai (I): nustatoma, jei IRQ pertraukimas išjungtas.
- Neigiamas (N): nustatomas, jei nustatytas reikšmingiausias rezultato bitas.
- Nulis (Z): nustatomas, jei rezultatas lygus nuliui.
- Perpildymas (V): nustatomas, jei paskutinio rezultato skaičiavimo metu buvo perpildymas.
- Carry (C): nustatomas, jei paskutinio rezultato skaičiavimo metu buvo perkėlimas iš 7 bito.

Intel 4040

Intel 4040 neturi specialaus vėliavų registro kaip Motorola 6800, tačiau naudoja požymių bitus. Naudoja “Carry flag” (C).

- **Koks buvo kiekvienos architektūros duomenų plotis?**

Motorola 6800 duomenų plotis 8 bitai.

Intel 4040 duomenų plotis 4 bitai.

- **Koks buvo kiekvienos sistemos atminties išdėstymas?**

Motorola 6800

“Program, data and stack memories occupy the same memory space. The total addressable memory size is 64 KB.

Program memory - program can be located anywhere in memory. Jump and subroutine call instructions can be used to jump anywhere in memory. Conditional and unconditional branches are limited to memory addresses positioned no farther than -125 - +129 bytes from the branch instruction.

Data memory - data can be anywhere in memory space.

Stack memory - stack can be placed anywhere in memory space.

Reserved memory locations:

- FFF8h - FFF9h: Pointer to IRQ interrupt-processing routine.
- FFFAh - FFFBh: Pointer to software interrupt-processing routine.
- FFFCh - FFFDh: Pointer to NMI interrupt-processing routine.
- FFFEh - FFFFh: Pointer to RESET handling code.
- Some memory addresses may be reserved for memory mapped I/O as the processor doesn't have hardware I/O capability.” (CPU-World, 2024)

Intel 4040

“**Program memory** size is 8 KB organized as two 4 KB memory banks. At any point of time the processor works with one 4 KB bank, when necessary the processor can switch to different bank using new DB0 and DB1 instructions. Like in Intel 4004, all conditional instructions in the 4040 work within currently

selected ROM (256 bytes). Unconditional jump and jump to subroutine instructions can be used to jump to any address within currently selected bank.

Data memory size is 640 bytes. RAM access is done in the same way as access to I/O ports. First, a SRC instruction is used to tell the processor what memory address to access, and successive WRM or RDM writes accumulator data to memory or reads data into accumulator. Data memory is separate from program memory.

Stack is 7-level deep. Stack is separate from program memory and data memory.” (CPU-World, 2024)

Ar adresų erdvė ištisinė ar ji buvo suskirstyta?

Motorola 6800 atmintis yra ištisinė, nes “Program, data and stack memories occupy the same memory space.” (CPU-World, 2024)

Intel 4040 adresų erdvė yra segmentuota, nes “Program memory size is 8 KB organized as two 4 KB memory banks.” (CPU-World, 2024).

Koks buvo (efektyvus) adresų plotis?

Motorola 6800 efektyvus adresų plotis buvo 16 bitų, o Intel 4040 12 bitų.

Koks buvo maksimalus įmanomas atminties kiekis kiekvienoje sistemoje?

Motorola 6800: “The total addressable memory size is 64 KB.” (CPU-World, 2024).

Intel 4040: “Program memory size is 8 KB organized as two 4 KB memory banks. <...> Data memory size is 640 bytes.” (CPU-World, 2024).

• Ar buvo palaikoma virtualioji atmintis?

Nei Motorola 6800 nei Intel 4040 nepalaiko virtualios atminties. Abi turėjo fizinę atmintį.

- **Kokia buvo kiekvienos sistemos architektūros komandų sistema (ISA)?**

Motorola 6800 architektūros komandų sistema buvo CISC, taip pat ir Intel 4040.

Kiek mašinos komandų turėjo kiekviena sistema?

Motorola 6800 turėjo 72 instrukcijas, o Intel 4040 - 60.

Kokios buvo instrukcijų (komandų) klasės?

Motorola 6800:

- Duomenų perkėlimo instrukcijos.
- Aritmetinės: sudėtis, atimtis, neigimas, inkrementuojančios, dekrementuojančios, lyginimo operacijos.
- Loginės: AND, OR, XOR, komplementavimas, poslinkio/rotacijos instrukcijos.
- Kontrolės instrukcijos: sąlyginiai ir nesąlyginiai šuoliai, “subrout” iškvietai ir grįžimai.
- Kitos: sąlygų vėliavėlių išvalymas/nustatymas, bitų testai, programinės įrangos nutraukimas ir kt.

Intel 4040:

- Duomenų perkėlimo instrukcijos.
- Aritmetinės: pridėti, atimti, didinti, mažinti.
- Loginės: pasukimo, AND ir OR.
- Valdymo perdavimo: (ribojamas iki esamo ROM), besąlyginis, iškviečiama paprogramė ir grįžtama iš paprogramės.
- Įvesties/išvesties instrukcijos.
- Su pertraukimu susijusios instrukcijos – sustabdyti, įjungti, išjungti ir grįžti iš paprogramės.
- Kita – pernešimo vėliavėlės operacijos, dešimtainis koregavimas, banko pasirinkimas ROM ir indeksų registras ir kt.

Kokius instrukcijų formatus palaikė kiekviena architektūra? Pateikti 8-16 pavyzdžius.

Motorola 6800 instrukcijos (ne visos):

MNEMONO	SYNTAX	MODE	BYTES	CODE	CYCLES	C	Z	S	O	A _c	I	SYMBOLIC OPERATION	DESCRIPTION		
ABA	ABA	ACC	1	\$1B	2	x	x	x	x	x	-	[A] ← [A] + [B]	Add B to A		
ADC	ADC A #data8	IMM	2	\$89	2	x	x	x	x	x	-	[A] ← [A] + data8 + C	Add contents of Memory + C to Accumulator		
	ADC A addr8	DIR	2	\$99	3							[A] ← [A] + [addr8] + C			
	ADC A data8,X	IDX	2	\$A9	5							[A] ← [A] + [data8 + [X]] + C			
	ADC A addr16	EXT	3	\$B9	4							[A] ← [A] + [addr16] + C			
	ADC B #data8	IMM	2	\$C9	2							[B] ← [B] + data8 + C			
	ADC B addr8	DIR	2	\$D9	3							[B] ← [B] + [addr8] + C			
	ADC B data8,X	IDX	2	\$E9	5							[B] ← [B] + [data8 + [X]] + C			
ADD	ADC B addr16	EXT	3	\$F9	4	[B] ← [B] + [addr16] + C	x	x	x	x	-	[A] ← [A] + data8	Add Memory contents to the Accumulator		
	ADD A #data8	IMM	2	\$8B	2	[A] ← [A] + data8									
	ADD A addr8	DIR	2	\$9B	3	[A] ← [A] + [addr8]									
	ADD A data8,X	IDX	2	\$AB	5	[A] ← [A] + [data8 + [X]]									
	ADD A addr16	EXT	3	\$BB	4	[A] ← [A] + [addr16]									
	ADD B #data8	IMM	2	\$CB	2	[B] ← [B] + data8									
	ADD B addr8	DIR	2	\$DB	3	[B] ← [B] + [addr8]									
AND	ADD B data8,X	IDX	2	\$EB	5	[B] ← [B] + [data8 + [X]]	-	x	x	0	-	[B] ← [B] + [addr16]	Memory contents AND the Accumulator to the Accumulator		
	ADD B addr16	EXT	3	\$FB	4	[B] ← [B] + [addr16]									
	AND A #data8	IMM	2	\$84	2	[A] ← [A] ∧ data8									
	AND A addr8	DIR	2	\$94	3	[A] ← [A] ∧ [addr8]									
	AND A data8,X	IDX	2	\$A4	5	[A] ← [A] ∧ [data8 + [X]]									
	AND A addr16	EXT	3	\$B4	4	[A] ← [A] ∧ [addr16]									
	AND B #data8	IMM	2	\$C4	2	[B] ← [B] ∧ data8									
ASL	AND B addr8	DIR	2	\$D4	3	[B] ← [B] ∧ [addr8]	x	x	x	x	-	-	C ← 76543210 ← 0	Arithmetic Shift Left. Bit 0 is 0 (multiplying by two)	
	AND B data8,X	IDX	2	\$E4	5	[B] ← [B] ∧ [data8 + [X]]									
	AND B addr16	EXT	3	\$F4	4	[B] ← [B] ∧ [addr16]									
	ASL A	ACC	1	\$48	2	C ← 76543210 ← 0									Arithmetic Shift Left. Bit 0 is 0 (multiplying by two)
	ASL B	ACC	1	\$58	2										
	ASL data8,X	IDX	2	\$68	7										
	ASL addr16	EXT	3	\$78	6										

ASR	ASR A	ACC	1	\$47	2	x	x	x	x	-	-	7[6]5[4]3[2]1[0] → C	Arithmetic Shift Right. Bit 7 stays the same.
	ASR B	ACC	1	\$57	2								
	ASR data8,X	IDX	2	\$67	7								
	ASR addr16	EXT	3	\$77	6								
BCC	BCC disp	REL	2	\$24	4	-	-	-	-	-	-	$(C == 0) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if carry clear
BCS	BCS disp	REL	2	\$25	4	-	-	-	-	-	-	$(C == 1) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if carry set
BEQ	BEQ disp	REL	2	\$27	4	-	-	-	-	-	-	$(Z == 1) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if equal to zero
BGE	BGE disp	REL	2	\$2C	4	-	-	-	-	-	-	$(S \vee O == 0) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if greater than or equal to zero
BGT	BGT disp	REL	2	\$2E	4	-	-	-	-	-	-	$(Z \vee (S \vee O) == 0) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if greater than zero
BHI	BHI disp	REL	2	\$22	4	-	-	-	-	-	-	$(C \vee Z == 0) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if Accumulator contents higher than comparand
BIT	BIT A #data8	IMM	2	\$85	2	-	x	x	0	-	-	$[A] \wedge \text{data8}$	Memory contents AND the Accumulator, but only Status register is affected.
	BIT A addr8	DIR	2	\$95	3							$[A] \wedge [\text{addr8}]$	
	BIT A data8,X	IDX	2	\$A5	5							$[A] \wedge [\text{data8} + [X]]$	
	BIT A addr16	EXT	3	\$B5	4							$[A] \wedge [\text{addr16}]$	
	BIT B #data8	IMM	2	\$C5	2							$[B] \wedge \text{data8}$	
	BIT B addr8	DIR	2	\$D5	3							$[B] \wedge [\text{addr8}]$	
	BIT B data8,X	IDX	2	\$E5	5							$[B] \wedge [\text{data8} + [X]]$	
	BIT B addr16	EXT	3	\$F5	4							$[B] \wedge [\text{addr16}]$	
BLE	BLE disp	REL	2	\$2F	4	-	-	-	-	-	-	$(Z \vee (S \vee O) == 1) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if less than or equal to zero
BLS	BLS disp	REL	2	\$23	4	-	-	-	-	-	-	$(C \vee Z == 1) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if Accumulator contents less than or same as comparand
BLT	BLT disp	REL	2	\$2D	4	-	-	-	-	-	-	$(S \vee O == 1) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if less than zero
BMI	BMI disp	REL	2	\$2B	4	-	-	-	-	-	-	$(S == 1) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if minus
BNE	BNE disp	REL	2	\$26	4	-	-	-	-	-	-	$(Z == 0) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if not equal to zero
BPL	BPL disp	REL	2	\$2A	4	-	-	-	-	-	-	$(S == 0) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if plus
BRA	BRA disp	REL	2	\$20	4	-	-	-	-	-	-	$[PC] \leftarrow [PC] + \text{disp} + 2$	Unconditional branch relative to present Program Counter contents.
BSR	BSR disp	REL	2	\$8D	8	-	-	-	-	-	-	$[[SP]] \leftarrow [PC(LO)],$ $[[SP] - 1] \leftarrow [PC(HI)],$ $[SP] \leftarrow [SP] - 2,$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Unconditional branch to subroutine located relative to present Program Counter contents.
BVC	BVC disp	REL	2	\$28	4	-	-	-	-	-	-	$(O == 0) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if overflow clear
BVS	BVS disp	REL	2	\$29	4	-	-	-	-	-	-	$(O == 1) ?$ $[PC] \leftarrow [PC] + \text{disp} + 2$	Branch if overflow set
CBA	CBA	INH	1	\$11	2	x	x	x	x	-	-	$[A] - [B]$	Compare contents of Accumulators A and B. Only the Status register is affected.

Intel 4040 instrukcijos (ne visos):

INSTRUCTION SET SUMMARY

The instruction set of the 4040 and 4004 (CPU) are shown below. The following section will describe each instruction in detail.
[Those instructions preceded by an asterisk (*) are 2 word instructions that occupy 2 successive locations in ROM]

MACHINE INSTRUCTIONS (Logic 1 = Low Voltage = Negative Voltage (V_{DD}); Logic 0 = High Voltage = (V_{SS}))

MNEMONIC	OPR D ₃ D ₂ D ₁ D ₀	OPA D ₃ D ₂ D ₁ D ₀	DESCRIPTION OF OPERATION
NOP	0 0 0 0	0 0 0 0	No operation.
*JCN	0 0 0 1 A ₂ A ₂ A ₂ A ₂	C ₁ C ₂ C ₃ C ₄ A ₁ A ₁ A ₁ A ₁	Jump to ROM address A ₂ A ₂ A ₂ A ₂ A ₁ A ₁ A ₁ A ₁ (within the same ROM that contains this JCN instruction) if condition C ₁ C ₂ C ₃ C ₄ ⁽¹⁾ is true, otherwise skip (go to the next instruction in sequence).
*FIM	0 0 1 0 D ₂ D ₂ D ₂ D ₂	R ₁ R ₂ R ₃ 0 ₄ D ₁ D ₁ D ₁ D ₁	Fetch immediate (direct) form ROM Data D ₂ D ₂ D ₂ D ₂ D ₁ D ₁ D ₁ D ₁ to index register pair location RRR. ⁽²⁾
SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X ₂ and X ₃ time in the Instruction Cycle.
FIN	0 0 1 1	R R R 0	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.
JIN	0 0 1 1	R R R 1	Jump indirect. Send contents of register pair RRR out as an address at A ₁ and A ₂ time in the Instruction Cycle.
*JUN	0 1 0 0 A ₂ A ₂ A ₂ A ₂	A ₃ A ₃ A ₃ A ₃ A ₁ A ₁ A ₁ A ₁	Jump unconditional to ROM address A ₃ A ₃ A ₃ A ₃ A ₂ A ₂ A ₂ A ₂ A ₁ A ₁ A ₁ A ₁ .
*JMS	0 1 0 1 A ₂ A ₂ A ₂ A ₂	A ₃ A ₃ A ₃ A ₃ A ₁ A ₁ A ₁ A ₁	Jump to subroutine ROM address A ₃ A ₃ A ₃ A ₃ A ₂ A ₂ A ₂ A ₂ A ₁ A ₁ A ₁ A ₁ , save old address. (Up 1 level in stack)
INC	0 1 1 0	R R R R	Increment contents of register RRRR. ⁽³⁾
*ISZ	0 1 1 1 A ₂ A ₂ A ₂ A ₂	R R R R A ₁ A ₁ A ₁ A ₁	Increment contents of register RRRR. Go to ROM address A ₂ A ₂ A ₂ A ₂ A ₁ A ₁ A ₁ A ₁ (within the same ROM that contains this ISZ instruction) if result is not 0, otherwise skip (go to the next instruction in sequence).
ADD	1 0 0 0	R R R R	Add contents of register RRRR to accumulator with carry.
SUB	1 0 0 1	R R R R	Subtract contents of register RRRR from accumulator with borrow.
LD	1 0 1 0	R R R R	Load contents of register RRRR to accumulator.
XCH	1 0 1 1	R R R R	Exchange contents of index register RRRR and accumulator.
BBL	1 1 0 0	D D D D	Branch back (down 1 level in stack) and load data DDDD to accumulator.
LDM	1 1 0 1	D D D D	Load data DDDD to accumulator.

4040 ONLY INSTRUCTIONS

MNEMONIC	OPR D ₃ D ₂ D ₁ D ₀	OPA D ₃ D ₂ D ₁ D ₀	DESCRIPTION OF OPERATION
HLT	0 0 0 0	0 0 0 1	Halt - inhibit program counter and data buffers.
BBS	0 0 0 0	0 0 1 0	Branch Back from Interrupt and restore the previous SRC. The program counter and send register control are restored to their pre-interrupt value.
LCR	0 0 0 0	0 0 1 1	The contents of the COMMAND REGISTER are transferred to the accumulator.
OR4	0 0 0 0	0 1 0 0	The 4 bit contents of index register #4 are logically "OR-ed" with the accumulator.
OR5	0 0 0 0	0 1 0 1	The 4 bit contents of index register #5 are logically "OR-ed" with the accumulator.
AN6	0 0 0 0	0 1 1 0	The 4 bit contents of index register #6 are logically "AND-ed" with the accumulator.
AN7	0 0 0 0	0 1 1 1	The 4 bit contents of index register #7 are logically "AND-ed" with the accumulator.
DB0	0 0 0 0	1 0 0 0	DESIGNATE ROM BANK 0. CM-ROM ₀ becomes enabled.
DB1	0 0 0 0	1 0 0 1	DESIGNATE ROM BANK 1. CM-ROM ₁ becomes enabled.
SB0	0 0 0 0	1 0 1 0	SELECT INDEX REGISTER BANK 0. The index registers 0 - 7.
SB1	0 0 0 0	1 0 1 1	SELECT INDEX REGISTER BANK 1. The index registers 0* - 7*.
EIN	0 0 0 0	1 1 0 0	ENABLE INTERRUPT.
DIN	0 0 0 0	1 1 0 1	DISABLE INTERRUPT.
RPM	0 0 0 0	1 1 1 0	READ PROGRAM MEMORY.

Motorola 6800 ir Intel 4040 panašios komandos: ADD, SUB.

Motorola 6800 turi CMP komandą, o Intel 4040 ne ir pan.

- Kokius adresavimo būdus palaikė kiekviena architektūra?

Motorola 6800 adresavimo būdai:

- Numanomas – duomenų reikšmė/duomenų adresas netiesiogiai susiejamas su instrukcija.
- Akumuliatorius – instrukcija reiškia, kad duomenys yra vienas iš akumuliatoriaus registų.
- “Immediate” (nedelsiantis) – instrukcijoje pateikti 8 bitų arba 16 bitų duomenys.
- Tiesioginis – instrukcijoje pateiktas vieno baido operandas nurodo atminties adresą nuliniam puslapyje (0000h – 00FFh), kuriame yra duomenys.
- Išplėstinis – instrukcijoje pateiktas dviejų baitų operandas nurodo atminties adresą, kuriame yra duomenys.
- Santykinis – vienas baito poslinkis pridedamas prie kitos komandos adreso (programos skaitiklio registro turinys + 2). Poslinkis yra ženklas su ženklu nuo -127 iki +127.
- Indeksuotas – prie IX registro turinio pridedamas vieno baito operandas, gauta 16 bitų reikšmė yra rodyklė į atmintį, kurioje yra duomenys.

Intel 4040 adresavimo būdai:

- Registro (4 bitai).
- Tiesioginė atmintis.
- Registro netiesiogiai. Registras netiesiogiai gali pasiekti duomenis tik dabartinėje ROM.
- Nedelsiantis (“immediate”) (4 ir 8 bitų duomenys).

Adresavimo būdai panašūs dėl nedelsiančio adresavimo (“immediate”). Taip pat dėl tiesioginio adresavimo.

Skiriasi dėl akumuliatoriaus, išplėstinio, adresavimo, santykinio, indeksuoto būdo, kurį turi Motorola 6800, bet neturi Intel 4040.

O Intel 4040 turi registro netiesioginį būdą, kurio neturi Motorola 6800.

- **Kokios buvo kiekvienos architektūros I/O galimybės?**

Motorola 6800 neturi I/O “portų”, o Intel 4040: 16 4 bitų įvesties ir 16 4 bitų išvesties.

- **Ar buvo palaikomi pertraukimai? Kuo pertraukimų mechanizmai buvo panašūs, kuo jie skyrėsi abiejose architektūrose?**

Motorola 6800 pertraukimai:

IRQ – maskuojamas pertraukimas. Įvykus pertraukimui programų skaitiklis, indeksų registras, akumulatoriai ir sąlygų kodų registrai yra išsaugomi steke, kiti pertraukimai išjungiami ir procesorius peršoka į atminties vietos adresą, kurios adresas yra saugomas atmintyje FFF8h - FFF9h. Norėdami grįžti iš pertraukimo, apdorojimo rutina turėtų naudoti RTI nurodymus. Šį pertraukimą galima įjungti / išjungti naudojant CLI/SEI instrukcijas.

NMI – neužmaskuojamas pertraukimas. Įvykus pertraukimui programų skaitiklis, indeksų registras, akumulatoriai ir sąlygų kodų registrai yra išsaugomi steke, kiti pertraukimai išjungiami ir procesorius peršoka į atminties vietos adresą, kurios atmintyje saugomas FFFCh - FFFDh. Norėdami grįžti iš pertraukimo, apdorojimo rutina turėtų naudoti RTI nurodymus. Šio pertraukimo negalima išjungti.

SWI – programinės įrangos pertraukimas. Šį pertraukimą galima iškviesti tik iš programos. Įvykus pertraukimui procesorius išsaugo programų skaitiklį, indeksų registrą, kaupiklius ir sąlygų kodų registrus, išjungia tolesnius pertraukimus ir peršoka į atminties vietos adresą, kurio adresas yra saugomas atmintyje FFFAh - FFFBh. Norėdami grįžti iš pertraukos, apdorojimo rutina turėtų naudoti RTI instrukcijas. Šio pertraukimo negalima išjungti.

Intel 4040 pertraukimai:

Procesorius turi vieną maskuojamą pertraukimą. Pertraukimą galima išjungti arba įjungti naudojant DIN ir EIN instrukcijas. Kai įvyksta pertraukimas, programos skaitiklio ir siuntimo registro (SRC) turinys išsaugomas. Norėdami grįžti iš pertraukos, pertraukimo apdorojimo kodas turi vykdyti BBS instrukcijas.

Motorola 6800 turi 3 pertraukimus, o Intel 4040 tik vieną.

Abi architektūros naudoja maskuotą pertraukimą, kai jis įvyksta procesorius išsaugo duomenis svarbius. Abiejose yra vykdomas specialus nurodymas, kad procesorius galėtų grįžti prie darbo, tačiau Motorola 6800 naudoja RTI nurodymus, o Intel 4040 naudoja BBS nurodymus.

Motorola 6800 išsaugo registrus: programos skaitiklį, indeksų registrą, akumuliatorių, sąlygų kodų registrą, o Intel 4040 išsaugo tik programos skaitiklį ir siuntimo registrą (SRC).

- **Kokius duomenų tipus palaikė kiekviena architektūra aparatūros lygyje?**

Motorola 6800 galėjo dirbti su 8 bitų sveikais skaičiais, o Intel 4040 su 4 bitų sveikais skaičiais.

Ar buvo palaikoma fiksuoto kablelio, slankiojo kablelio aritmetika?

Motorola 6800 nepalaikė tiesioginės fiksuoto kablelio aritmetikos. Nepalaikė ir slankiojo kablelio aritmetikos, tačiau slankiojo kablelio operacijos galėjo būti atliekamos programiškai, naudojant atitinkamus algoritmus ir operacijas su sveikaisiais skaičiais.

Intel 4040 taip pat nepalaikė fiksuoto kablelio aritmetikos. Kaip ir 6800, Intel 4040 nebuvo specialiai sukurta slankiojo kablelio aritmetikai, ir tokie skaičiavimai būtų turėję būti įgyvendinti programiškai.

Ar sveikieji skaičiai buvo koduojami kaip ženklas-dydis, kaip vieneto papildinys (atvirkštinis kodas), dvejetainis papildomas kodas? Kokius kitus „egzotiškus“ duomenų tipus palaikė?

Motorola 6800 palaikė sveikųjų skaičių operacijas, tiek su ženklu, tiek be ženklo. Tokių skaičių kodavimas: ženklas-dydis (sign-magnitude) kodavimas arba, dažniausiai, dvejetainis papildinys (two's complement) su 8 bitų pločiu, todėl nebuvo tiesiogiai naudojami tokie metodai kaip vieneto papildinys.

Intel 4040 palaikė sveikųjų skaičių operacijas, ir dažniausiai buvo naudojamas dvejetainis papildinys (two's complement) kaip kodavimo metodas, kaip ir Motorola 6800.

Taigi: nei Motorola 6800, nei Intel 4040 nepalaikė tiesiogiai slankiojo kablelio ar fiksuoto kablelio aritmetikos. Abi architektūros naudojo paprastus sveikuosius skaičius ir dažniausiai dvejeta papildinį.

Egzotiškų duomenų tipų palaikymas, kaip dešimtainiai ar kompleksiniai skaičiai, buvo įgyvendintas tik per programinius metodus (pvz., naudojant BCD arba kuriant atitinkamas struktūras ir algoritmus). Tiesiogiai nepalaikė tokių tipų.

- **Sistemos greitaveika**

Motorola 6800 dažnis buvo 1MHz iki 2MHz, o Intel 4040 buvo 500kHz iki 740kHz. Intel darė nuo 625000 iki 92 5000 taktų per sekundę.

Intel 4040 reikia 1 arba 2 mašininių ciklų kiekvienai komandai įvykdyti, tai apytiksliai 62 kIPS esant 740 kHz.

Vidutinė Intel 4040 greitaveika 1175 32 bitų skaičių sudėjimui. Per sekundę, ~850 μs.

Motorola 6800 pvz.: instrukcija LDA A #FF užtrunka 2 ciklus.

Motorola 6800 buvo našesnė nei Intel 4040.

- **Ar architektūros naudojo spartinančią atmintį? Jei taip, kokio dydžio?**

Nei Motorola 6800, nei intel 4040 nenaudojo spartinančios atminties.

- **Taikymo sritys**

Motorola 6800 tapo pagrindu keliems ankstyviesiems asmeniniams kompiuteriams bei įterptinėms sistemoms. Tokios įmonės kaip „Apple“, „Atari“ ir „Commodore“ pasirinko 6800 savo produktams.

Asmeniniai kompiuteriai: Apple I ir II: Pirmieji Apple kompiuteriai naudojo 6800 architektūrą, atverdami kelią asmeninių kompiuterių revoliucijai. Ypač Apple II tapo vienu sėkmingiausių namų kompiuterių, pasižymintis pažangiomis grafikos ir garso galimybėmis.

Atari konsolės: 1977 metais išleistas Atari 2600 naudojo 6800 procesorių žaidimų galimybėms, prisidedant prie namų žaidimų populiarumo.

Commodore PET: Commodore PET buvo vienas pirmųjų „all-in-one“ asmeninių kompiuterių, veikiantis su 6800 procesoriumi. Jo naudotojo patogus

dizainas ir integruotas monitorius padarė jį populiarią švietimo įstaigose ir tarp entuziastų.

Įterptinės sistemos

6800 buvo plačiai naudojamas įterptinėse sistemose dėl savo mažos kainos ir efektyvumo. Taikymo sritys apėmė:

Automobilių valdymo sistemos: Daugelis ankstyvųjų automobilių valdymo sistemų naudojo 6800 užduotims, tokioms kaip variklio valdymas ir prietaisų skydelio rodymas, leidžiant pasiekti pažangą automobilių technologijose.

Vartojimo elektronika: 6800 buvo integruotas į įvairius vartojimo įrenginius, įskaitant mikrobangų krosneles, skalbimo mašinas ir namų prietaisus, leisdamas automatizuoti ir valdyti šiuos įrenginius.

Pramonės automatizacija: 6800 buvo naudojamas programuojamuose loginiai valdikliuose (PLC) ir kitose pramonės taikymo srityse, leisdamas padidinti efektyvumą ir automatizavimą gamybos procesuose.

Palikimas ir poveikis

Motorola 6800 poveikis yra platesnis nei jo tiesioginės taikymo sritys. Jis turėjo įtakos vėlesnių mikroprocesorių ir kompiuterio architektūrų kūrimui.

Intel 4040 buvo naudojamas daugelyje ankstyvųjų vaizdo žaidimų ir žaidimų mašinų, tokių kaip Bailey shuffleboard. Buvo MCS-40 mikroschemų rinkinio dalis ir buvo naudojamas kalkuliatoriuose bei įterptinėse taikymo srityse.

Naudojamas kai kuriuose ankstyvuose žaidimų mašinose, pvz.: pinball žaidimai ir kai kurias arkadines žaidimų sistemas. Buvo integruotas ir į įvairius prietaisus, kaip valdymo sistemos ar ekranų valdiklis.

• Programinė įranga

Kriterijus	Motorola 6800	Intel 4040
Programinės įrangos kiekis	Daugiau programinės įrangos dėl platesnio naudojimo ir palaikymo	Ribotas programinės įrangos kiekis dėl trumpesnio naudojimo
Prieinamumas	Vis dar prieinami	Rečiau prieinama

Naudojimo sritys	Įterptosios sistemos, pramoniniai valdikliai, asmeniniai kompiuteriai (pvz., Altair 680)	Ankstyvieji įterptosios sistemos projektai, mokymo priemonės
Kompiliatoriai	Pascal, BASIC, C kompiliatoriai buvo prieinami	Labai ribota - daugiausiai assembleris
Programavimo įrankiai	- Assembleriai: Motorola M6800 Assembler, ASM68	- Assembleriai: Intel 4040 Assembler
	- Derintojai: Motorola EXORciser derintojas	- Pagrindinių derintojų mažai (daugiau naudojamas rankiniu būdu)
	- Emuliatoriai: Galimi retro sistemų emuliatoriai	- Riboti emuliatoriai, skirti istorinėms rekonstrukcijoms
Programinės bibliotekos	Daugiau bibliotekų skirtų įterptosioms sistemoms ir periferijoms	Ribotos, dauguma buvo projektuojama specifinei aparatinei įrangai
Populiarumas	Labiau paplitęs dėl technologinių galimybių	Mažiau populiarus, greitai pakeistas modernesnėmis Intel serijomis

Šaltiniai

Wikipedia, 2024. Motorola 6800. [online] Prieinama:

https://en.wikipedia.org/wiki/Motorola_6800

Wikipedia, 2024. Intel 4040. [online] Prieinama:

https://en.wikipedia.org/wiki/Intel_4040.

NXP, 2024. SOT129-4. [online] Prieinama:

<https://www.nxp.com/packages/SOT129-4> .

Wikipedia, 2024. Dual in-line package. [online] Prieinama:

https://en.wikipedia.org/wiki/Dual_in-line_package .

Phinalabs, 2024. 24-Pin DIP IC Socket - Wide 0.6. [online] Prieinama:

<https://phinalabs.com/product/24-pin-dip-ic-socket-wide-0-6/> .

Amicus, 2024. Product Information. [online] Prieinama:

https://amicus.com.sg/index.php?route=product/product&product_id=897 .

Waveshare, 2024. DIP24 Aries Black. [online] Prieinama:

<https://www.waveshare.com/dip24-aries-black.htm>.

Waveshare, 2024. DIP-40 Pin ZIF Socket Green. [online] Prieinama:

<https://www.waveshare.com/dip-40-pin-zif-socket-green.htm>.

ChatGPT, 2024. Link 1. [online] Prieinama:

<https://chatgpt.com/share/675c4fa2-c028-800d-96eb-36332a598baf> .

Quora, 2024. What do you know about the 6800 microprocessor? [online]

Prieinama: https://www.quora.com/What-do-you-know-about-the-6800-microprocessor?top_ans=78436562 .

CPU-World, 2024. Intel 4040. [online] Prieinama: <https://www.cpu-world.com/Arch/4040.html> .

Wikipedia, 2024. Intel 4004. [online] Prieinama:

https://en.wikipedia.org/wiki/Intel_4004 .

Intel4004, 2024. QA4004. [online] Prieinama:

<http://www.intel4004.com/qa4004.htm>.

CPU-World, 2024. Motorola 6800. [online] Prieinama: <https://www.cpu-world.com/Arch/6800.html>.

CPU-World, 2024. Intel 4040. [online] Prieinama: <https://www.cpu-world.com/Arch/4040.html> .

Google, 2024. Google Translate. [online] Prieinama: <https://translate.google.com/?sl=en&tl=lt&op=translate> .

Wikipedia, 2024. Carry flag. [online] Prieinama: https://en.wikipedia.org/wiki/Carry_flag .

Bou.lt, 2024. Computer Non-RISC. [online] Prieinama: <https://www.bou.lt/theory/computerNonRisc.pdf>

Quora, 2024. Does Intel use CISC or RISC architecture in today's processors? [online] Prieinama: <https://www.quora.com/Does-Intel-use-CISC-architecture-or-RISC-architecture-in-todays-processors>

8bit-era.cz, 2024. Motorola 6800. [online] Prieinama: <http://www.8bit-era.cz/6800.html>

Deramp, 2024. Motorola 6800 Instruction Set. [online] Prieinama: [https://deramp.com/downloads/mfe_archive/050-Component%20Specifications/Motorola/CPUs/The%20Motorola%206800%20Instruction%20Set%20\(Byte\)\(January%201978\).pdf](https://deramp.com/downloads/mfe_archive/050-Component%20Specifications/Motorola/CPUs/The%20Motorola%206800%20Instruction%20Set%20(Byte)(January%201978).pdf)

Pastraiser, 2024. Intel 4040. [online] Prieinama: <https://pastraiser.com/cpu/i4040/i4040.html>

ChatGPT, 2024. Link 2. [online] Prieinama: <https://chatgpt.com/share/6760793d-1e40-800c-93bc-8c4bc8c497ff>

The Chip Letter, 2024. Motorola's Pioneering 8-bit 6800 Origins. [online] Prieinama: <https://thechipletter.substack.com/p/motorolas-pioneering-8-bit-6800-origins>

The Code Academy, 2024. Motorola 6800 Microprocessor. [online] Prieinama: <https://thecodeacademy.co.uk/docs/electronics/microprocessors/the-motorola-6800-microprocessor-a-comprehensive-overview/>

Wikichip, 2024. Intel 4040. [online] Prieinama: <https://en.wikichip.org/wiki/intel/mcs-40/4040>

Scribd, 2024. Intel 4004 and Intel 4040. [online] Prieinama: <https://www.scribd.com/presentation/416387297/Intel-4004-and-Intel-4040>

Wikipedia, 2024. Intel 4040. [online] Prieinama:

https://en.wikipedia.org/wiki/Intel_4040

ChatGPT, 2024. Link 3. [online] Prieinama:

<https://chatgpt.com/share/6761bb0e-adc4-800c-80ae-7e114b7f22cc>