

EN-CHIPS MICROCONTROLLERS

En-chips microcontrollers, såsom Intels MCS-51 serie som beskrivs i denna artikel, håller mer och mer på att hitta vägen in i fler och fler industriella styrsystem. De erbjuder anmärkningsvärt låg chip-count såväl som enkel programmering när datorn tilldelas en enda uppgift.

För många applikationer har användningen av en speciell microcontroller uppenbara fördelar över mikroprocessorer av standardtyp som 6502 eller Z80. Detta beror huvudsakligen på att microcontrollern har skräddarsydda funktioner för att utföra endast de nödvändiga funktionerna. In- och utgångslinjerna är till exempel tillgängliga direkt på chipet, vilket gör att man inte behöver komplexa uppställningar med individuella I/O-chips, interface, buffrar och all nödvändig hårdvara för adress- och datautkoppling. Ofta har microcontrollern även en programmerbar timer/räknare och har dessutom utbyggda instruktioner för bitmanipulation. Typerna 8051 och 8052 är en-chips microcontrollers från Intels omfattande MCS-51 serie. Under senare år har de funnit ett ökat antal användningar inom industriella styrsystem. Denna introduktion är avsedd att göra er bekanta med 8051, 8052 och 8052AH-BASIC, vilka är mångsidiga och kraftfulla microcontrollers som tillåter design och uppbyggnad av kompakt, mikroprocessorbaserad utrustning för ett brett område applikationer.

MCS-51 microcontrollers

Tekniska egenskaper:

- 8-bitars CPU optimerad för styreapplikationer
- Utbyggd Booleansk process (enkelt-bits logik)
- 32 dubbelriktade och individuellt adresserbara I/O-linjer
- 128 eller 256 bytes on-chip data RAM
- 2 eller 3, 16-bitars timer/räknare
- Programmerbar, full duplex UART
- 5 eller 6 källor för interruptstruktur med 2 prioritetsnivåer
- On-chip klocksäfflar
- 4 eller 8Kbytes on-chip programminne (8751 & 8752: EPROM)
- 64 Kbyte adressutrymme för programminne
- 64 Kbyte adressutrymme för dataminne
- 111 instruktioner (64 enkelsyklar)
- Decimale och hexadecimala operationer
- 8 Kbyte BASIC-tolk (8052AH-BASIC)
- Inbyggd EPROM-programmerare under BASIC-kontroll (8052AH-BASIC)
- Speciella BASIC-kommandon för I/O, räknare och serieinterface (8052AH-BASIC)

MCS-51 familjen

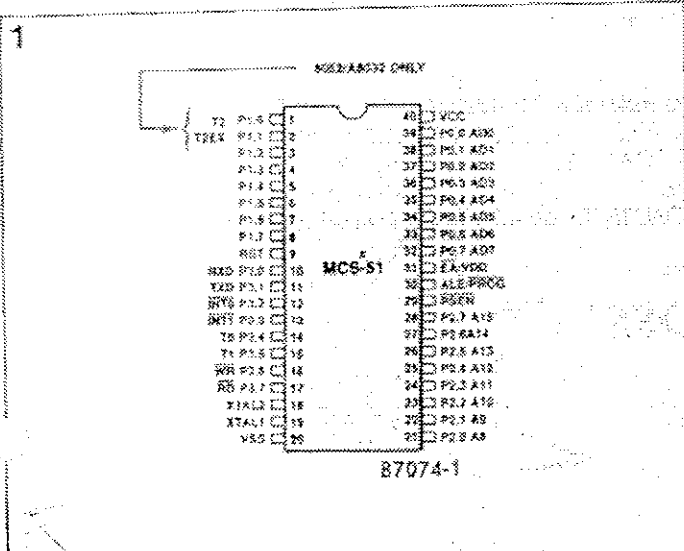
Intels MCS-51 familj av en-chips microcontrollers består av dem som listas i tabell 1. Det skall noteras att typindelningen 8051 ofta refererar till MCS-51 familjen i sin helhet. Benkonfigurationen och funktionsrepresentationen hos några av chipen i MCS-51 familjen visas i figurerna 1 respektive 2. 8051 är en controller med

on-chip ROM. I stället tar denna alla instruktioner från ett externt minne och är därför idealisk att använda när man skriver och testar mjukvara för 8051. På samma sätt är 8051AH och 8052AH de ROM-lösa versionerna av 8051AH respektive 8052AH medan motsvarande EPROM-versioner identifieras som 8751 och 8752. Mjukvaruskydd kan man få genom att EPROM-innehållet görs otillgängligt för yttre läsapparater. En separat produkt, 8052AH-BASIC, är i huvudsak en 8052AH med en kraftfull och snabb BASIC-tolk programmerad i det inbyggda ROM:et. Detta microcontrollerchip är speciellt intressant för engångsapplikationer och detta återkommer vi till mer detaljerat.

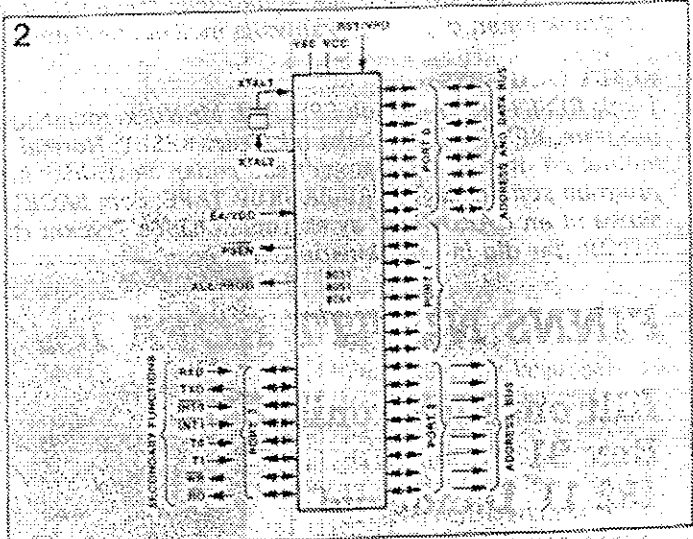
Minnesorganisation

Det viktigaste hos minnesstrukturen hos chipset i MCS-51 familjen syns i figur 3. I kortet är minnet indelat i 2 block om 64Kbytes. En block är programminnet, det andra dataminnet. De lägre 4 K-

4 Kbytes on-chip, maskprogrammerbar, ROM, 8052AH, ett HMOS II chip, är en förbättrat, helt kompatibel version av 8051. Det är viktigt att notera att 8051, 8051AH och 8052AH är maskprogrammerbara chips. Detta betyder att de endast finns tillgängliga i stora kvantiteter eftersom det inbyggda programminnet endast kan laddas av tillverkaren. 8051 skiljer sig emellertid från 8051 genom att inte ha något



Figur 1. Benkonfiguration för chipset i MCS-51 serien.



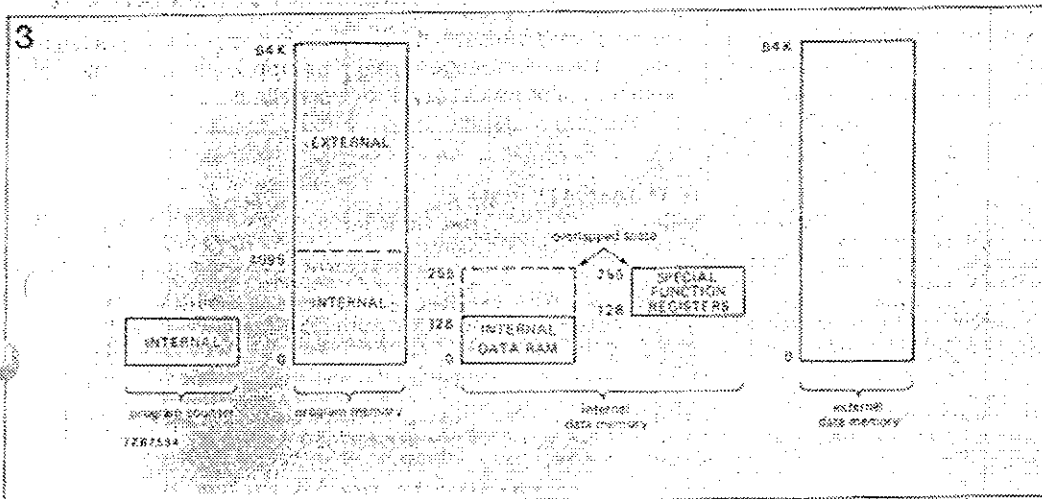
Figur 2. Visar portlinjefunktionerna när kontrollern opererar med externt minne.

Wahlstr. elektronik 02-84 62 10

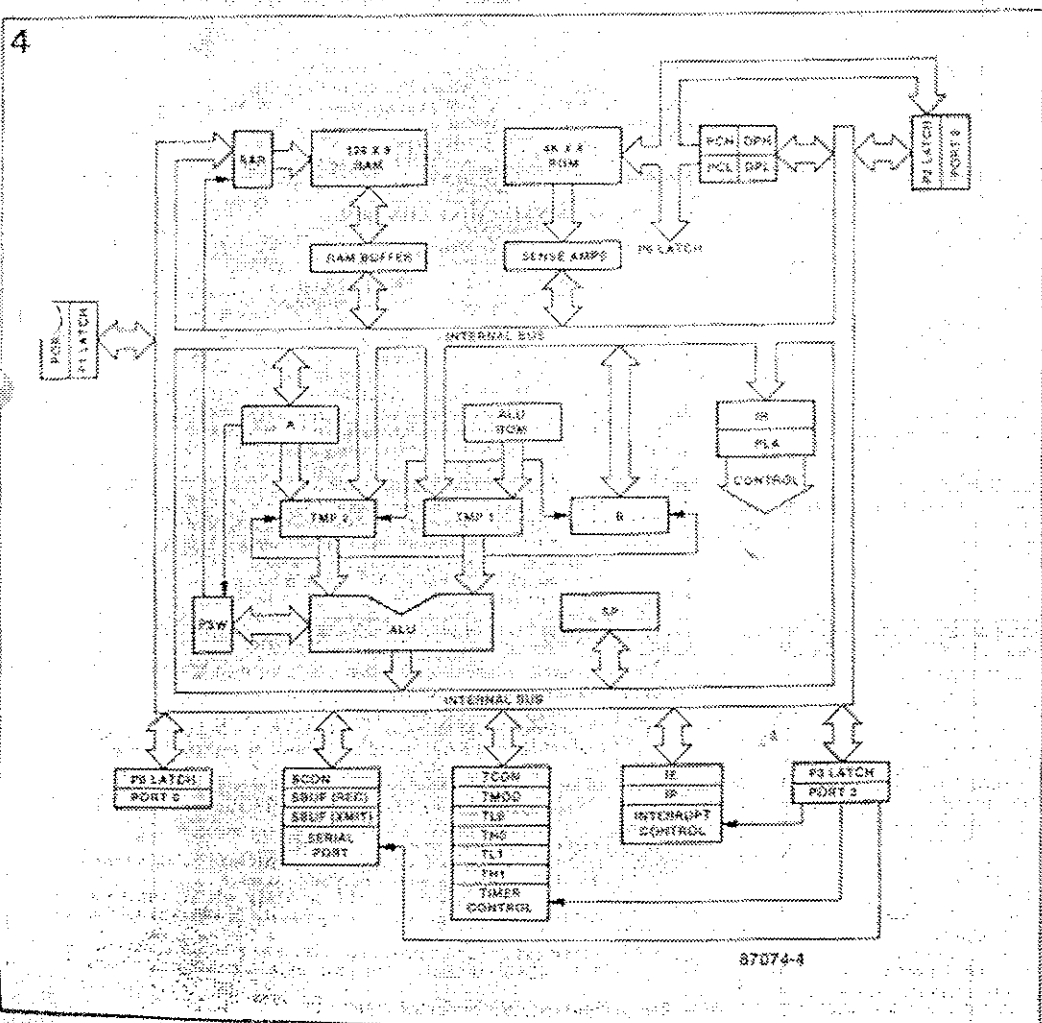
Table 1

Device	Internal Memory		Timers Event Counters	Interrupts
	Program	Data		
8052AH	8K x 8 ROM	256 x 8 RAM	3 x 16-Bit	6
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8051	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8032AH	none	256 x 8 RAM	3 x 16-Bit	5
8031AH	none	128 x 8 RAM	2 x 16-Bit	5
8031	none	128 x 8 RAM	2 x 16-Bit	5
8751H	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5
8751H-12	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5

Tabell 1. Microcontrollern i MCS-51 familjen.



Figur 3. Minnesstrukturen hos MCS-51.



Figur 4. Arkitekturen hos MCS-51.

lar 8 Kbyten hos programminnet är on-chip ROM. I ROM-lösa controllers är allt programminne externt och kan därför vara RAM eller (EP)ROM. Microcontrollers kan endast läsa från programminnet, vilket håller sekvensen av exekveringsbara processorinstruktioner i form av maskinkod. Dataminnet används för lagring av variabler, delresultat i beräkningar, behandlade dataord, uppslagstabeller och liknande. Upp till 64Kbytes av yttre ROM, RAM eller EPROM kan adresseras i det externa datautrymmet. Ett antal speciella registerfunktioner är lokaliserad i en separat 128 eller 256 byte RAM-area. Huvudregistren är ackumulatoren, register B (för multiplikation och division), statusregistret, stackpekaren, datapekaren (2x8 bitar eller 1x16 bitar) port 0-3, det dubbla seriella sändar/mottagarregistret, upplägningsregister för den tredje räknaren (8052) samt kommandoregistrer för specialfunktioner (interrupts, RTC, seriell I/O).

Det flesta MCS-51 instruktionerna exekveras under en enkel maskincykel, dvs inom 12 klockperioder. När klockfrekvensen är 12 MHz varar en maskincykel μs så att processhastigheten hos ett MCS-51 chip är densamma som hos en 6802 CPU som körs med 2 MHz eller en Z80 vid 8 MHz. En noterbart aspekt hos microcontrollerna är deras möjlighet att manipulera färre än 8 bitar samtidigt vilket befriar programmeraren från en hög bit-maskering.

Bussar och portar

Den grundläggande yttre strukturen hos en controller i MCS-51 familjen visas i figur 4. Men kan se att där finns, teoretiskt, 4 dubbelriktade, 8 bitar breda portar. I praktiken är dessa endast tillgängliga när det interna minnet (ROM eller RAM) används. I alla andra fall fungerar portarna 0 och 2 som data- och adressbussar (se till figur 2) så att 2 portar återstår för I/O-applikationer. Port P2 ger adresssignalerna A15-A16, port P0 adresssignalerna A7-A0 och databitarna D7-D0, multiplexade med hjälp av ALE-pulsen (adress latch enable). Utgångarna RD och WR är helt enkelt utgångslinjer på port 3, internt programmerad för att ge pulserna för läs- och skrivoperationer i det yttre dataminnet.

lässtroben för yttre programmering signalen PSEN (program store enable). Det är intressant att notera att PSEN, såväl som ALE, aktiveras två gånger under varje maskin-cykel under exekvering av ett program i (EP)ROM eftersom två bytes fångas successivt under varje cykel. PSEN förblir inaktiv när maskinkoden lagras i det interna minnet och det tomma minnet (som det finns) är tomt. PSEN används normalt inte i konstruktioner med 8052AH-BASIC eftersom det interna ROM:et håller BASIC-toolken. Ingång EA (external address) aktiveras när pro-

cessorn skall läsa operationskoder från det yttre minnet hellre än från sin interna (EPROM, EA- ingången används också när man lägger på 21V-spänningen för programmering av det interna EPROM:et (8751, 8752).

Timers och räknare

Som vi redan sett i tabell 1 har 8052 en 16 bitars timer/räknare mer än 8051. Här nedan finns en överblick av de funktioner som utförs av timer/räknarblocken. I timermodi ökas registerinne-

hållet med 1 under varje maskin-cykel. Den maximala räknarhastigheten är därför 1/12 del av processorns klockhastighet. I räknarmod ökas registerinnehållet vid bakskanten av den signal som läggs till ingångarna T0, T1 eller T2 (den senare finns bara på 8052) Den maximala räknarhastigheten är 1/24 del av processorns klockhastighet. Räknar/timers 0 och 1 har 4 programmerbara mod, inklusive 8 och 16 bitars operation och automatiskt laddning av förinställt värde. Timer/räknare 1 kan programmeras att fungera som bsdgenerator för det asynkro-

na serieinterfacet. Timer/räknare 2 (endast 8052) har 3 mod: 16-bitars automatiskt återladdningsbar räknare, 16-bitars infångningsräknare och baudrategenerator.

Serieporten

Samtliga microcontrollers i MCS-81 familjen har ett on-chip, dubbelriktat serieinterface (UART) kapabelt att sända och ta emot data samtidigt. En speciell databuffer finns för att den asynkrona mottagaren skall kunna snabbt upp kommunikationen med seriella periferenhetar.

Tabell 2 ARITHMETIC OPERATIONS

Mnemonic	Description	Byte	Cyc
ADD A,Rn	Add Register to Accumulator	1	1
ADDC A,Rn	Add direct byte to Accumulator	1	1
ADD A,@R1	Add indirect RAM to Accumulator	1	1
ADD A,#data	Add immediate data to Accumulator	1	1
ADDC A,Rn	Add register to Accumulator with Carry	1	1
ADDC A,dirct	Add direct byte to A with Carry flag	1	1
ADDC A,@R1	Add indirect RAM to A with Carry flag	1	1
ADDC A,#data	Add immediate data to A with Carry flag	1	1
SUBB A,Rn	Subtract Register from A with Borrow	1	1
SUBB A,dirct	Subtract direct byte from A with Borrow	1	1
SUBB A,@R1	Subtract indirect RAM from A with Borrow	1	1
SUBB A,#data	Subtract immediate data from A with Borrow	1	1
INC A	Increment Accumulator	1	1
INC Rn	Increment register	1	1
INC dirct	Increment direct byte	1	1
INC @R1	Increment indirect RAM	1	1
DEC A	Decrement Accumulator	1	1
DEC Rn	Decrement register	1	1
DEC dirct	Decrement direct byte	1	1
DEC @R1	Decrement indirect RAM	1	1
INC DPTR	Increment Data Pointer	2	2
MUL AB	Multiply A x B	1	4
DIV AB	Divide A by B	1	4
DA A	Decimal Adjust Accumulator	1	1

LOGICAL OPERATIONS

Mnemonic	Description	Byte	Cyc
ANL A,Rn	AND register to Accumulator	1	1
ANL A,dirct	AND direct byte to Accumulator	1	1
ANL A,@R1	AND indirect RAM to Accumulator	1	1
ANL A,#data	AND immediate data to Accumulator	1	1
ANL dirct,A	AND Accumulator to direct byte	1	1
ANL @R1,A	AND immediate data to direct byte	1	1
ORL A,Rn	OR register to Accumulator	1	1
ORL A,dirct	OR direct byte to Accumulator	1	1
ORL A,@R1	OR indirect RAM to Accumulator	1	1
ORL A,#data	OR immediate data to Accumulator	1	1
ORL dirct,A	OR Accumulator to direct byte	1	1
ORL @R1,A	OR immediate data to direct byte	1	1
XRL A,Rn	Exclusive OR register to Accumulator	1	1
XRL A,dirct	Exclusive OR direct byte to Accumulator	1	1
XRL A,@R1	Exclusive OR indirect RAM to A	1	1
XRL A,#data	Exclusive OR immediate data to A	1	1
XRL dirct,A	Exclusive OR Accumulator to direct byte	1	1
XRL @R1,A	Exclusive OR immediate data to direct byte	1	1
CLR A	Clear Accumulator	1	1
CPL A	Complement Accumulator	1	1
RL A	Rotate Accumulator Left	1	1
RLC A	Rotate A Left through Carry flag	1	1
RR A	Rotate Accumulator Right	1	1
RRC A	Rotate A Right through Carry flag	1	1
SWAP A	Swap nibbles within the Accumulator	1	1

DATA TRANSFER

Mnemonic	Description	Byte	Cyc
MOV A,Rn	Move register to Accumulator	1	1
MOV A,dirct	Move direct byte to Accumulator	1	1
MOV A,@R1	Move indirect RAM to Accumulator	1	1
MOV A,#data	Move immediate data to Accumulator	1	1
MOV Rn,A	Move Accumulator to register	1	1
MOV Rn,dirct	Move direct byte to register	1	1
MOV Rn,#data	Move immediate data to register	1	1
MOV dirct,A	Move Accumulator to direct byte	1	1
MOV dirct,Rn	Move register to direct byte	1	1
MOV dirct,@R1	Move direct byte to indirect RAM	1	1
MOV @R1,A	Move Accumulator to indirect RAM	1	1
MOV @R1,dirct	Move direct byte to indirect RAM	1	1
MOV @R1,#data	Move immediate data to indirect RAM	1	1
MOV DPTR,#data16	Load Data Pointer with a 16-bit constant	2	2

DATA TRANSFER (cont)

Mnemonic	Description	Byte	Cyc
MOVC A,@DPTR	Move Code byte relative to DPTR to A	1	2
MOVC A,@APC	Move Code byte relative to PC to A	1	2
MOVA A,Rn	Move Internal RAM to direct address to A	1	2
MOVXA A,@DPTR	Move External RAM (data address) to A	1	2
MOVXA A,@R1	Move into External RAM (data address)	1	2
MOVXA A,@DPTR,A	Move into External RAM (data address)	1	2
PUSH dirct	Push direct byte onto stack	1	2
POP dirct	Pop direct byte from stack	1	2
XCH A,Rn	Exchange register with Accumulator	1	1
XCH A,dirct	Exchange direct byte with Accumulator	1	1
XCH A,@R1	Exchange indirect RAM with A	1	1
XCHD A,@R1	Exchange low order 8 bits ind. RAM w. A	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic	Description	Byte	Cyc
CLR C	Clear Carry flag	1	1
CLR dirct	Clear direct bit	1	1
SETB C	Set Carry flag	1	1
SETB dirct	Set direct bit	1	1
CPL C	Complement Carry flag	1	1
CPL dirct	Complement direct bit	1	1
ANL C,dirct	AND direct bit to Carry flag	1	1
ANL C,C	AND complement of direct bit to Carry	1	1
ORL C,dirct	OR direct bit to Carry flag	1	1
ORL C,C	OR complement of direct bit to Carry	1	1
MOV C,dirct	Move direct bit to Carry flag	1	1
MOV C,C	Move Carry flag to direct bit	1	1

PROGRAM AND MACHINE CONTROL

Mnemonic	Description	Byte	Cyc
ACALL addr16	Absolute Subroutine Call	2	2
LCALL addr16	Long Subroutine Call	3	2
RET	Return from sub routine	1	1
RETI	Return from interrupt	1	1
AJMP addr11	Absolute Jump	2	2
LJMP addr16	Long Jump	3	2
SJMP rel	Short Jump (relative addr)	1	2
JMP @A+DPTR	Jump indirect relative to the DPTR	1	2
JZ rel	Jump if Accumulator = Zero	2	2
JNZ rel	Jump if Accumulator Not Zero	2	2
JC rel	Jump if Carry flag set	2	2
JNC rel	Jump if No Carry flag	2	2
J1 rel	Jump if direct bit set	1	2
J0 rel	Jump if direct bit not set	1	2
JNB bit,rel	Jump if direct bit Not set	1	2
JBC bit,rel	Jump if direct bit set & Clear bit	1	2
CJNE A,dirct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE A,#data,rel	Compare immediate to A & Jump if Not Equal	3	2
CJNE Rn,#data,rel	Compare immediate to Rn & Jump if Not Equal	3	2
CJNE @R1,#data,rel	Compare immediate to indirect RAM & Jump if Not Equal	3	2
DJNZ Rn,rel	Decrement register & Jump if Not Zero	3	2
DJNZ dirct,rel	Decrement direct & Jump if Not Zero	3	2
NOF	No operation	1	1

Notes on data addressing modes:

- Rn Working register, R0-R7
- dirct 128 internal RAM locations, any I/O port control or status register
- @R1 Indirect internal RAM location addressed by register R0 or R1
- #data 8-bit constant included in instruction
- addr16 16-bit constant included as bytes 1 & 2 of instruction
- rel 128 software flags, any I/O port control or status bit

Notes on program addressing modes:

- addr16 Decimal address for LCALL & LJMP may be anywhere within the 64 Kbyte program memory address space
- addr11 Destination address for ACALL & AJMP will be within the same 2 Kbyte page of program memory as the first byte of the following instruction
- rel SJMP and all conditional jumps include an 8-bit offset byte. Range is +127, -128 bytes relative to first byte of following instruction.

All mnemonics copyrighted © Intel Corporation 1979

Tabell 2. De kompletta instruktionerna för 8051/8052 serien av microcontrollers från Intel.

Serieporten kan programmeras att operera i 1 av 4-mod med inbak styrd baudrate och dataformat. Alla vanliga baudrates upp till 19 200 kan väljas, såväl som klockhastigheter upp till 1 MHz för användning i nät och kommunikationssystem med flera processorer. Välet av klockhastighet utförs med hjälp av timer/räknarna.

Interrupts och instruktioner

8051 och 8052 känner igen 5 respektive 8 interruptkällor: INT0 och INT1 (programmerbara för extern nivå eller pulskantsdetektering), timer/räknarna 0 och 1 samt 2 i 8052) och serieporten. Verr programmering med 2 prioritetnivåer är helt oberoende av hårdvarans konfiguration, och en av de 8 eller 6 interruptkällorna kan tilldelas en individuell vektor (adresspekare). En interrupt på "så sätt" tas not hoppas processorn till den relevanta servicerutinen efter att den lagrat innehållet i "programräknaren" i stacken. Vanliga instruktioner hos controller i MCS-51 familjen visas i tabell 3.

8052AH-BASIC

8052AH-BASIC är utan tvekan den mest intressanta i MCS-51 familjen som erbjuder ett elegant och effektivt sätt att skriva program för datorbaserade system. Intel har laddat ROM:et med 8Kbytes i detta chip med en full BASIC och medan en ytterligare timer/räknare, T3, tillåter split baudrateoperation av serierinterfacet. BASIC:ens instruktioner (version 1.1) listas i tabell 3. Om man tittar lite närmare ser man att det finns specialkommandon inkluderade som inte kräver assemblyspråk. Ett mångfald faster och utlösningar blev slutsatsen att 8052AH-BASIC passar utmärkt i olika applikationer i anslutning till fjärrstyrd datalogging, inmatning och mätning samt seriell processstyrning. Carven är naturligtvis långt från perfekt med maskinspråk har fördelen av att kunna programmeras med ett "normalt" C. Söskning och redigering av program är också betydelsefullt och erfarna program-

merare kan lära sig processorns kapacitet ganska fort med ett minimum av hårdvara. Dessutom känns det tryggt att veta att Intel kan tillhandahålla en 200-sidors referensmanual för 8052AH-BASIC programmerare.

Vissa speciella egenskaper hos 8052AH-BASIC förtjänar speciell uppmärksamhet. Controllern kan programmeras i stort sett vilken typ av EPROM som helst om rätt programmeringsspänning läggs till ben EA. Både den "gamla" programmeringsstandard (30 ns per adress) och flera versioner av den interaktiva algoritmen supportas, så de 14 sista uttrycken under Commands i tabell 3. RAM, ROM och XFER är kommandon för manipulering och överföring av minnesblock. Mångsidigheten och programmeringskraften hos BASIC-processorn är uppenbar om man ser antalet speciella Statements BAUD sätter dataöverföringen på den tidigare nämnda extra serierkanalen. CALL tillåter uppkallandet av maskinspråksrutiner från BASIC. CLOCK styr en realtidsklocka och CLEARs initialiserar datastacken vilken primärt används för utbyte av maskinspråkparametrar eller för lagring av lokala variabler. Ytterligare intressanta möjligheter erbjuds av ON-TIME, ett kommando som tillåter generering av interrupt vid en förprogrammerad tid. ONEX1 för hopp till en subrutin som följer en INT1 interruptförfrågan (intressant eftersom externa och timerinterrupts kan hanteras via BASIC). PH0 och PH1 för skrivning i hexadecimalt format. PUSH och POP för stackmanipulering i BASIC och slutligen PWM för generering av en pulsbreddsmodulerad signal. Kommandona inkluderar också ett antal användbara instruktioner för uppkallning av tolkrutiner direkt från BASIC: UI, UI0, UI1 och UO0.

Den tredje kolumnen i tabell 3. Operators listar inte bara de välkända aritmetiska och logiska operationerna utan också ett antal speciella funktionsoperatorer, börjande med CBY0. Dessa ope-

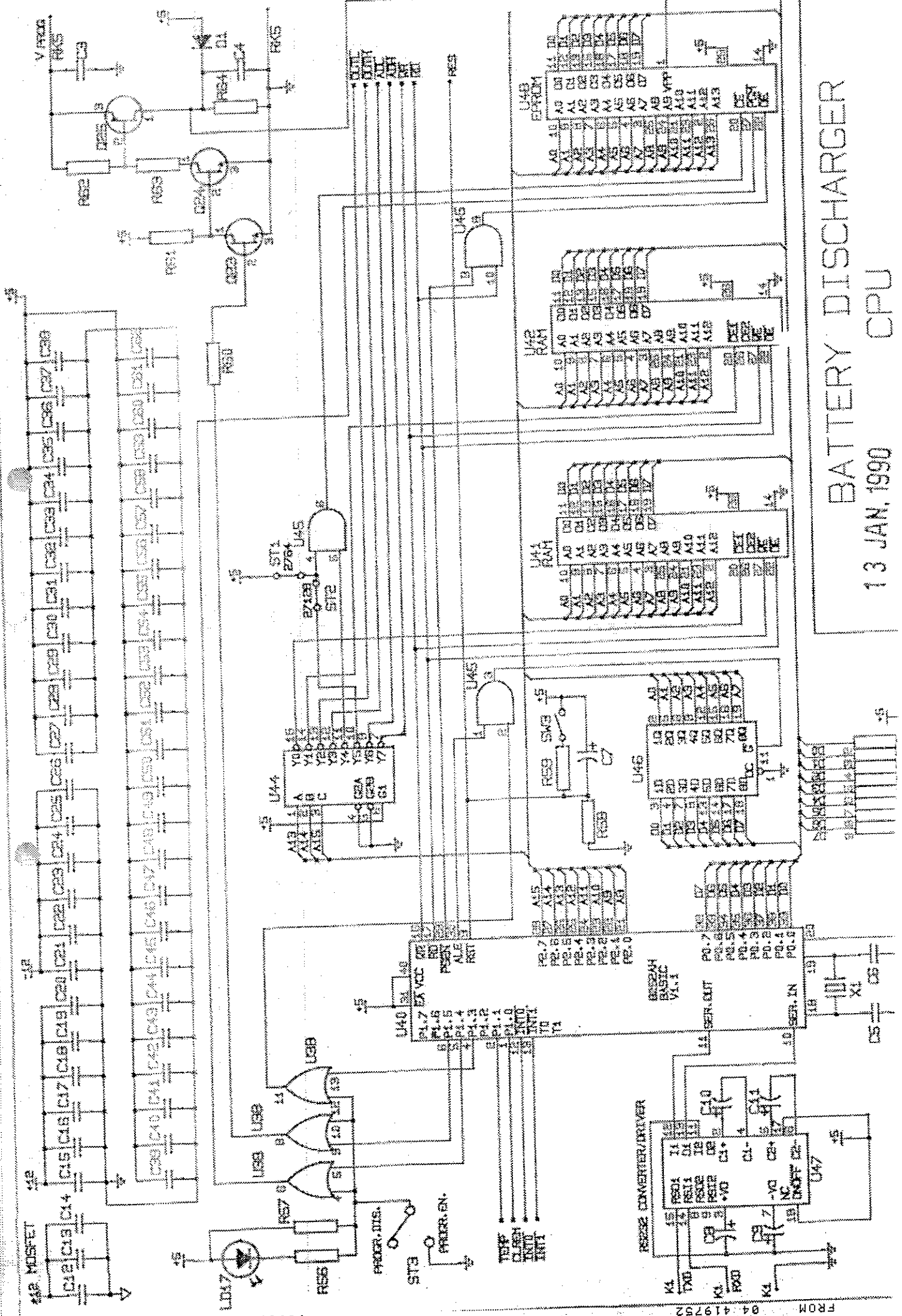
eratorer erbjuder direkt kontroll över I/O-linjerna och minneslokaliseringarna. CBY0 och DBY0 till exempel, ger läs/skrivåtkomst till det interna programrespektive dataminnet. GET tillåter ett tecken att läsas från serierinterfacet. De övriga specialoperatorerna, upp till och inklusive TIMER2, är avsedda för läsning från och skrivning till det register som indikeras av den relevanta förkortningen. Operator XTAL förser processorn med information angående den använda klockfrekvensen. Detta behövs för att se till korrekt operation av realtidsklockan. Statusen hos minnet kan läsas vid alla tider tack vare operatorm TOP som returnerar den högsta tillgängliga minneslokaliseringen. LEN, för information om programlängden och FREE för antalet tillgängliga minneslokaliseringar. Något som är ganska märkligt är att 8052AH-BASIC tillåter BASIC-funktioner att kallas upp från assemblyspråk. Dessa funktioner inkluderar flyttalsberäkningar, komplexa aritmetiska operationer och ingångs/utgångsrutiner. Microcontrollerna i MCS-51 serien kommer att vara hjärtat i en serie projekt som för närvarande är under utveckling i Elektrors konstruktionsavdelning. Detaljer om tillgänglighet och programmering kommer så småningom.

Källa: *Embedded Controller Handbook*, Intel Corporation.

Table 3

COMMANDS	STATEMENTS	OPERATORS
Halt	BAND	ADD (+)
CONT	CALL	DIVIDE (/)
EXIT	CLEAR	EXPONENTIATION (**)
LIST*	CLEAR (SA)	MULTIPLY (*)
LIST	CLOCK (180)	SUBTRACT (-)
NEW	DATA	LOGICAL AND (AND)
NULL	READ	LOGICAL OR (OR)
RAM	RESTORE	LOGICAL X-OR (XOR)
ROM	DIM	LOGICAL NOT
XFER	DO WHILE	ABS
PROG	DO UNTIL	INT1
PROG1	END	SGN
PROG2	FOR-TO-STEP	SQR
PROG3	NEXT	RND
PROG4	GOSUB	LOG
PROG5	RETURN	EXP1
PROG6	GOTO	SIN1
FPROG1	ON-GOTO	CGE
FPROG2	ON-GOSUB	TAN1
FPROG3	IF-THEN-ELSE	ATN1
FPROG4	INPUT	
FPROG5	LET	ASC1
FPROG6	ONERR	CHR1
	ONEX1	CBY1
	ONTIME	DBY1
	PRINT	XBY1
	PRINT*	GET
	PRINT*	IE
	PH0	W
	PH1	PORT
	PH1*	PCON
	PHI	RCAP2
	PHI*	TCON
	PHI	TCON
	PGM	TMOV
	PUSH	TIME
	POP	TIMER0
	PWM	TIMER1
	ROM	TIMER2
	RETI	XTAL
	STOP	MTOP
	STRING	LEN
	UI (180)	FREE
	UI0 (180)	PI
	UI1	
	UO0	
	UO1	
	UO2	
	UO3	
	UO4	
	UO5	
	UO6	
	UO7	
	UO8	
	UO9	
	UO10	
	UO11	
	UO12	
	UO13	
	UO14	
	UO15	
	UO16	
	UO17	
	UO18	
	UO19	
	UO20	
	UO21	
	UO22	
	UO23	
	UO24	
	UO25	
	UO26	
	UO27	
	UO28	
	UO29	
	UO30	
	UO31	
	UO32	
	UO33	
	UO34	
	UO35	
	UO36	
	UO37	
	UO38	
	UO39	
	UO40	
	UO41	
	UO42	
	UO43	
	UO44	
	UO45	
	UO46	
	UO47	
	UO48	
	UO49	
	UO50	
	UO51	
	UO52	
	UO53	
	UO54	
	UO55	
	UO56	
	UO57	
	UO58	
	UO59	
	UO60	
	UO61	
	UO62	
	UO63	
	UO64	
	UO65	
	UO66	
	UO67	
	UO68	
	UO69	
	UO70	
	UO71	
	UO72	
	UO73	
	UO74	
	UO75	
	UO76	
	UO77	
	UO78	
	UO79	
	UO80	
	UO81	
	UO82	
	UO83	
	UO84	
	UO85	
	UO86	
	UO87	
	UO88	
	UO89	
	UO90	
	UO91	
	UO92	
	UO93	
	UO94	
	UO95	
	UO96	
	UO97	
	UO98	
	UO99	
	UO100	

Tabell 3. BASIC-instruktionerna hos 8052AH-BASIC inkluderar ett antal kraftfulla och effektiva kommandon, programsatser och operatorer.



BATTERY DISCHARGER CPU

13 JAN. 1990