بسمه تعالی

نام درس:

اصول میکرو

فهرست مطالب

- مقدمه ای بر میکرو کنترلرها
- معماری
- خلاصه دستورالعمل ها
- تایمر
- درگاه سریال
- وقفه ها

منابع

میکروکنترلر ۸۰۵۱

پدیدآورنده: اسکات مکنزی
مترجم: حمیدرضا رضایی نیا
پیمان دربندی آذر
نشر: باغانی
شابک: ۵-۲-۹۱۵۳۲-۹۶۴

میکروکنترلر ۸۰۵۱

پدیدآورنده: محمدعلی مزیدی
جانیس گیلیسپی مزیدی
مترجم: قدرت الله سپیدنام
نشر: باغانی
شابک: ۰-۰۰-۷۳۴۳-۹۶۴

ارزشیابی

پروژه اول	---------	۵ نمره
میان ترم	---------	۵ نمره
پروژه دوم	---------	۵ نمره
پایان ترم	۹۰/۰٤/۰۱	۵ نمره

۲۰ نمره

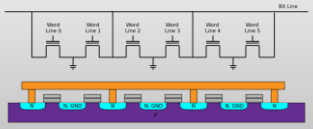
فصل اول:

مقدمه ای بر میکروکنترلرها

مقدمه

میکرو کنترلر چیست؟

- یک میکروکنترلر (که گاهی اوقات به اختصار به آن **µC**، **UC** و یا **MCU** نیز اطلاق می گردد) یک کامپیوتر کوچک در یک مدار مجتمع بوده که شامل هسته پردازنده، حافظه و لوازم جانبی ورودی / خروجی قابل برنامه ریزی است.
- علاوه بر مقدار کم معمول رم، حافظه برنامه در قالب فلش **NOR** و یا رام **OTP** نیز اغلب در تراشه گنجانده میشود.



هر سلول یک انتهای زمین شده دارد و انتهای دیگر آن به یک بیت لاین وصل می گردد.

به این آرایش به این دلیل فلش **NOR** می گویند که مانند یک گیت **NOR** عمل می کند یعنی در زمانی که از یکی خطوط کلمه **high** گردد ترانزیستور متناظر آن به گونه ای عمل می کند که خروجی خط بیت **low** گردد.

میکرو کنترلر چیست؟ (... ادامه)

- میکروکنترلر در محصولات و دستگاه هایی که به صورت خودکار کنترل می شوند مورد استفاده قرار می گیرد، از قبیل:

– سیستم های کنترل موتور خودرو – تجهیزات پزشکی کاشتنی – کنترل از راه دور

– ماشین های اداری – لوازم خانگی – اسباب بازی – و ...

- با کاهش اندازه و هزینه، میکروکنترلرها در مقایسه با طرح هایی که از ریز پردازنده، حافظه و ورودی/خروجی جداگانه استفاده می کنند، کنترل فرآیندها و دستگاه های دیجیتالی را مقرون به صرفه تر می سازند.

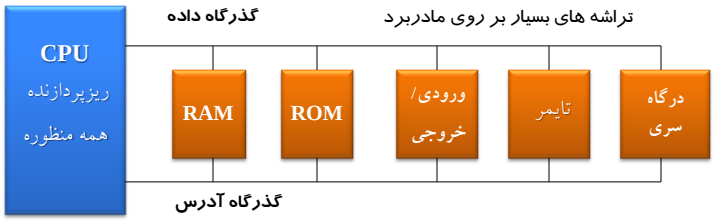
اجزای یک ریزپردازنده /**میکروکنترلر**

- **CPU**: واحد پردازش مرکزی
- **I/O**: ورودی/خروجی
- **گذرگاه**: گذرگاه آدرس و گذرگاه داده
- **حافظه**: رم و رام
- **تایمر**
- **وقفه**
- **درگاه سری**
- **درگاه موازی**

ریزپردازنده

ریزپردازنده همه منظوره

- **CPU** برای کامپیوترها
- معمولاً هیچ ورودی/خروجی و حافظه ای (رم و رام) بر روی تراشه پردازنده وجود ندارد.

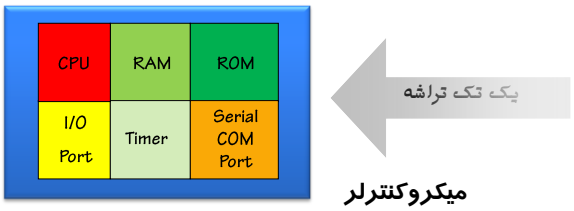


Intel's x86: 8086,8088,80386,80486, Pentium
Motorola's 680x0: 68000, 68010, 68020,68030,6040

میکروکنترلر

میکروکنترلر

- کامپیوتری با یک تراشه
- با حافظه و درگاه های ورودی و خروجی بر روی همان تراشه



میکروکنترلر

ریزپردازنده

- CPU، حافظه، ورودی/خروجی و تایمر از همدیگر مجزا هستند.
- طراح می تواند در مورد مقادیر حافظه و ورودی/خروجی تصمیم گیری نماید.
- گران است.
- تطبیق پذیر است.
- همه منظوره است.
- از توان پردازش بالایی برخوردار است.
- مصرف توان بالایی دارد.
- مجموعه دستورالعمل های آن متمرکز بر عملیات پردازش فشرده است.
- به طور معمول ۳۲ یا ۶۴ بیتی هستند.

میکرو کنترلر

- CPU، حافظه، ورودی/خروجی و تایمر همگی بر روی یک تراشه قرار دارند.
- مقادیر حافظه و ورودی/خروجی روی تراشه از قبل تثبیت شده اند.
- برای کاربردهایی که در آنها هزینه، توان مصرفی و فضا حیاتی است.
- تک منظوره است.
- توان پردازشی آن کم است.
- توان مصرفی آن پایین است.
- مجموعه دستورالعمل های آن متمرکز بر عملیات بر روی بیت ها و کاربردهای کنترلی است.
- به طور معمول ۸ یا ۱۶ بیتی هستند.

فهرست میکروکنترلرها

لیستی از نام های تجاری میکرو کنترلرهای رایج

1 AMCC
2 Altera
3 Analog Devices
4 Atmel
5 Charmed Labs
6 Cypress Semiconductor
7 Dallas Semiconductor
8 ELAN Microelectronics Corp.
9 Energy Micro AS
10 EPSON Semiconductor
11 Freescale Semiconductor
12 Fujitsu
13 Holtek
14 Infineon
15 Intel
16 Lattice Semiconductor
17 Microchip Technology
18 National Semiconductor
19 NEC
20 Parallax
21 NXP Semiconductors
22 Rabbit Semiconductor
23 Renesas Electronics
24 SiLabs
25 Silicon Motion
26 Sony
27 STMicroelectronics
28 Texas Instruments
29 Toshiba
30 Ubicom
31 Xemics
32 Xilinx
33 ZiLOG
34 Sortable table

فهرست میکروکنترلرها

AT89 series
(Intel 8051 architecture)

AT90
(AVR architecture)
- Attiny
- Atmega
- ATxmega

AT91SAM
(ARM architecture)

AVR32
(32-bit AVR architecture)

MARC4

8-bit
- MCS-48

8048 family
8035, 8038, 8039
8040, 8X42, 8X49
8050; X=0 or 7
- MCS-51

8051 family:
8X31, 8X32, 8X52
X=0, 3, or 7
- 8xC251

16-bit
- MCS-96

8096 family
8061Intel
- MCS-296

8-bit
- PIC10
- PIC12
- PIC16
  - PIC16F628
  - PICAXE
- PIC18

16-bit
- PIC24
- dsPIC

32-bit
- PIC32MX

TMS370
8-bit

MSP430
16-bit, Ultra-low-power

TMS320F28xx
32-bit

C2000
32-bit, Real-time control

Stellaris
32-bit, ARM Cortex-M3

TMS570
32-bit RISC

Older:
- Zilog Z8

8-bit architecture
- Zilog Z180

Z80 based micro

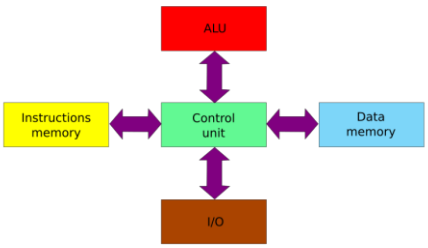
Newer:
- Zilog eZ8
- Zilog eZ80
- Zilog Z16

فصل دوم:

معماری ۸۰۵۱

Intel MCS-51

- **Intel MCS-51** سری میکرو کنترلر تک تراشه ای معماری هاروارد است که به وسیله اینتل در سال ۱۹۸۰ عرضه گردید.
- معماری هاروارد، معماری کامپیوتری است با حافظه ذخیره سازی فیزیکی، مسیر سیگنال داده و دستورالعمل جداگانه



- نسخه های اصلی اینتل در دهه ۱۹۸۰ و اوایل ۱۹۹۰ محبوب بوده اما امروزه تا حد زیادی توسط طیف گسترده ای از تجهیزات سریع تر تولید شده همساز با ۸۰۵۱ به وسیله بیش از ۲۰ کارخانه شامل :

Atmel, Infineon Technologies (سابق Siemens AG), Maxim Integrated Products, NXP (سابق Philips Semiconductor), Nuvoton (سابق Winbond), ST Microelectronics, Silicon Laboratories (سابقCygnal), Texas Instruments, Ramtron International, Silicon Storage Technology, Cypress Semiconductor.

جایگزین گردیده است.

- خانواده **MCS-51** اصلی اینتل با استفاده از تکنولوژی **NMOS** ایجاد گردیده بود اما در نسخه های بعدی که با حرف **C** در نام هایشان مشخص گردیده است (به عنوان مثال **80C51**) از تکنولوژی **CMOS** استفاده شد که نسبت به نوع **NMOS** از توان مصرفی پایین تری برخوردار بودند که آنها را برای تجهیزاتی که از باتری برای تغذیه استفاده می کنند مناسب تر ساخت

- ۸۰۵۱ اولین عضو این خانواده است که به صورت تجاری ارائه گردید که خلاصه مشخصات آن به شرح زیر است:

  – ۴ کیلوبایت ROM (۸ کیلوبایت برای 8052)
  - ۱۲۸ بایت RAM (۲۵۶ بایت برای 8052)
  - ۴ پورت ورودی- خروجی ۸ بیتی(PIO)
  - – دو تایمر ۱۶ بیتی (۳ تایمر برای 8052)
  - پورت سریال(Serial Interface)
  - امکان دسترسی به ۶۴ کیلوبایت حافظه بیرونی برای برنامه(Code)
  - امکان دسترسی به ۶۴ کیلوبایت حافظه بیرونی برای داده (Data)
  - امکان کار روی بیت (Boolean Processor)
  - ۲۱۰ بیت قابل آدرس دهی
  - توانایی انجام عملیات ضرب و تقسیم در کمتر از $4\mu s$

- قبل از ۸۰۵۱، ۸۰۸۴ در صفحه کلید اولین **IBM PC** مورد استفاده قرار گرفت که فشار کلیدها را به رشته اطلاعاتی تبدیل می کرد که به واحد اصلی کامپیوتر ارسال می شد. امروزه از ۸۰۸۴ و مشتقات آن هنوز برای مدل های ابتدایی صفحه کلید استفاده می گردد.

- ۸۰۳۱ یک نسخه خلاصه شده از ۸۰۵۱ بود که شامل هیچگونه حافظه برنامه داخلی **(ROM)** نبود. برای استفاده از این تراشه می بایست یک رام خارجی (حاوی برنامه ای که ۸۰۳۱ می بایست واکشی و اجرا نماید) به آن متصل می شد. یک تراشه ۸۰۵۱ می تواند به عنوان یک ۸۰۳۱ بدون رام فروخته شود به این صورت که رام داخلی ۸۰۵۱ با استفاده از حالت نرمال پین EA در یک طراحی مبتنی بر ۸۰۳۱ غیر فعال گردد. یک فروشنده ممکن است یک ۸۰۵۱ را به عنوان ۸۰۳۱ به دلایلی از قبیل عیب در رام ۸۰۵۱ یا به دلیل عرضه بیش از حد ۸۰۵۱ و عرضه کم ۸۰۳۱ بفروشد.

- ۸۰۵۲ نسخه پیشرفته ۸۰۵۱ بود که مشخصه برجسته آن ۲۵۶ بایت رم داخلی در برابر ۱۲۸ بایت، ۸ کیلو بایت رام در مقابل ۴ کیلو بایت و یک تایمر ۱۶ بیتی سوم بود. ۸۰۳۲ دارای خصوصیات مشابه با ۸۰۵۲ بود به جز حافظه برنامه رام داخلی.
- اینتل خط تولید **MCS-51** خودش را در مارس ۲۰۰۷ متوقف ساخت.

PART NUMBER	ON-CHIP CODE MEMORY	ON-CHIP DATA MEMORY	TIMERS
8051	4K ROM	128 bytes	2
8031	0K	128 bytes	2
8751	4K EPROM	128 bytes	2
8052	8K ROM	256 bytes	3
8032	0K	256 bytes	3
8752	8K EPROM	256 bytes	3

Table 1-5: Versions of 8051 From Atmel (All ROM Flash)

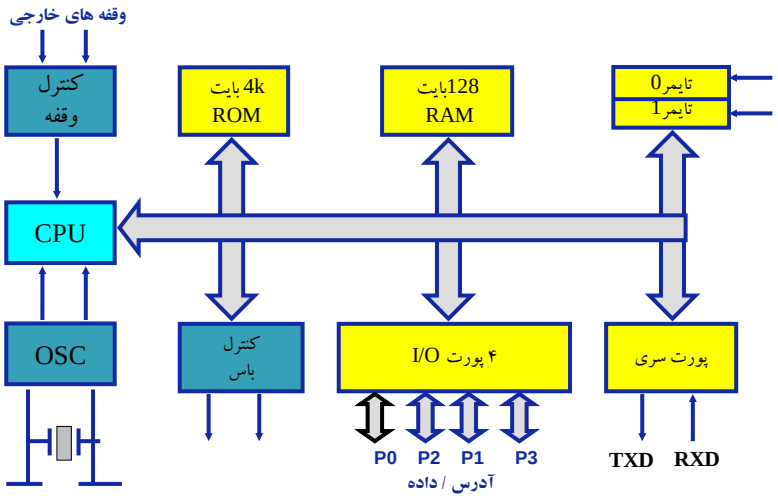
Part Number	ROM	RAM	I/O pins	Timer	Interrupt	V_{CC}
AT89C51	4K	128	32	2	6	5V
AT89LV51	4K	128	32	2	6	3V
AT89C1051	1K	64	15	1	3	3V
AT89C2051	2K	128	15	2	6	3V
AT89C52	8K	128	32	3	8	5V
AT89LV52	8K	128	32	3	8	3V

Table 1-6: Various Speeds of 8051 From Atmel

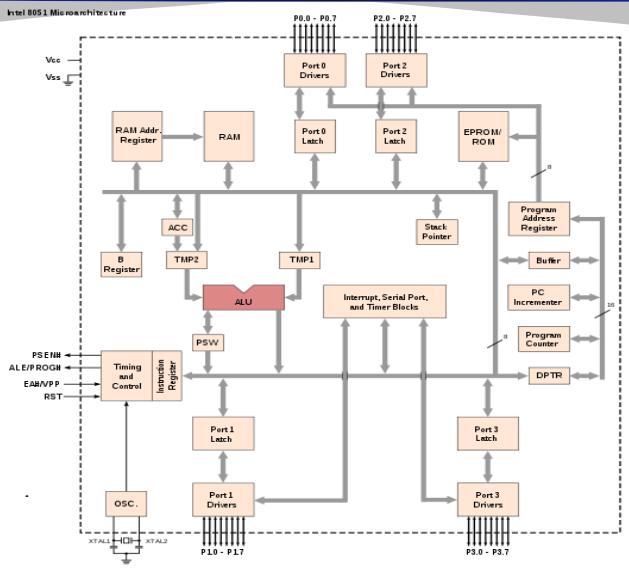
Part Number	Speed	Pins	Packaging	Use
AT89C51-12PC	12 MHz	40	DIP plastic	commercial
AT89C51-16PC	16 MHz	40	DIP plastic	commercial
AT89C51-20PC	20 MHz	40	DIP plastic	commercial

- AT= کارخانه **ATMEL** C= تکنولوژی **CMOS** LV= توان پایین

بلوک دیاگرام میکروکنترلر ۸۰۵۱ اینتل



معماری میکروکنترلر ۸۰۵۱



8051 Schematic Pin out

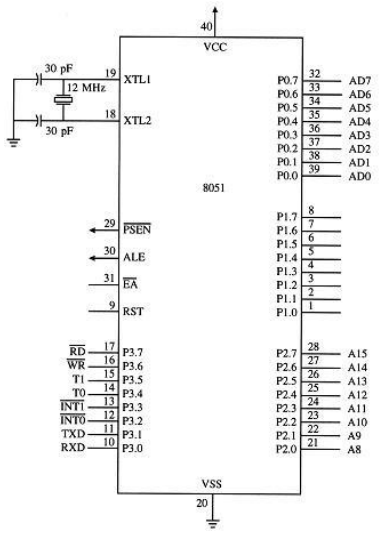
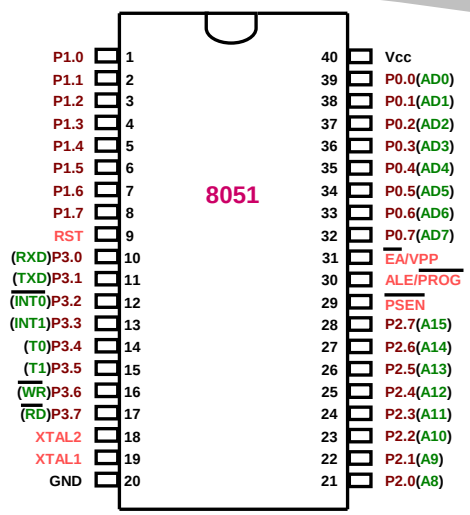


FIGURE 2–2
8051 pinouts

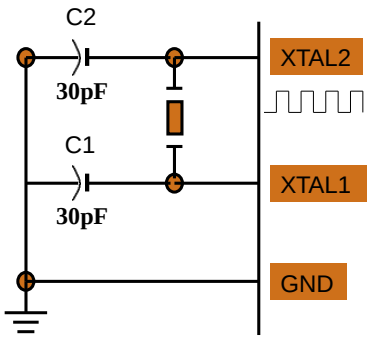
- ۳۲ پایه از ۴۰ پایه ۸۰۵۱ به عنوان خطوط درگاه ورودی/خروجی عمل می کنند.
- ۲۴ خط از این ۳۲ خط دو منظوره هستند .
- هر ۸ خط از یک درگاه می تواند به صورت یک واحد برای ارتباط با وسایل موازی مانند چاپگر به کار رود.
- هر خط خود به تنهایی می تواند با وسایل تک بیتی مانند کلید، **LED**، بلندگو و ... ارتباط برقرار نماید.



❑ اتصالات تغذیه

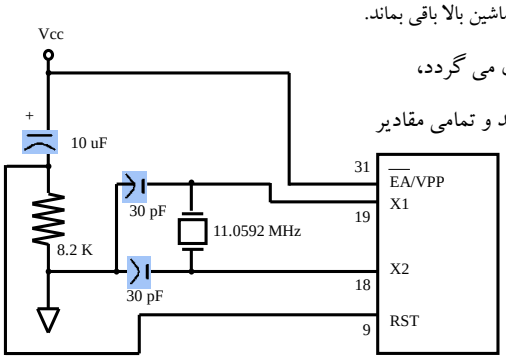
- **Vcc (pin 40) :**
  - Vcc ولتاژ تغذیه تراشه را فراهم می کند.
  - ولتاژ تغذیه مورد نیاز تراشه ۵+ ولت است.
- **GND (pin 20)**
  - زمین

❑ ورودی های نوسان ساز



- ۸۰۵۱ دارای یک نوسان ساز بر روی تراشه است و با یک کریستال که به پایه های ۱۸ و ۱۹ متصل است به راه می افتد.
- خازن های پایدارساز به صورت نشان داده شده در شکل مورد نیاز می باشد.
- نوسان ساز روی تراشه الزاماً نیاز به کریستال ندارد و می توان از یک منبع پالس **TTL** و اتصال آن به **XTAL1** و **XTAL2** استفاده نمود

- **RST （pin 9） : راه اندازی مجدد**
- یک پین ورودی است و در وضعیت بالا فعال می گردد (در شرایط عادی در وضعیت پایین است)
  - پالس بالا می بایست حداقل در دو چرخه ماشین بالا باقی بماند.
  - زمانی که یک پالس بالا به **RST** اعمال می گردد، میکرو کنترلر دوباره راه اندازی خواهد شد و تمامی مقادیر ثبات ها ناپدید خواهد شد.



- **EA / （pin 31） : دسترسی خارجی**
- هیچ رام روی تراشه ای در ۸۰۳۱ و ۸۰۳۲ وجود ندارد.
  - این پین برای اینکه مشخص گردد کد بر روی حافظه خارجی ذخیره می گردد به **GND** متصل می گردد.
  - **PSEN**/ و ALE برای رام خارجی مورد استفاده قرار می گیرد.
  - در ۸۰۵۱ پین EA/ به طور معمول به Vcc وصل می گردد.

- **PSEN/ （pin 29） : فعال ساز ذخیره برنامه**
- این یک پین خروجی است که حافظه برنامه خارجی را فعال می سازد.
- این پایه معمولاً به پایه **OE** یک **ROM** متصل می گردد تا خواندن بایت های برنامه از آن امکان پذیر شود.

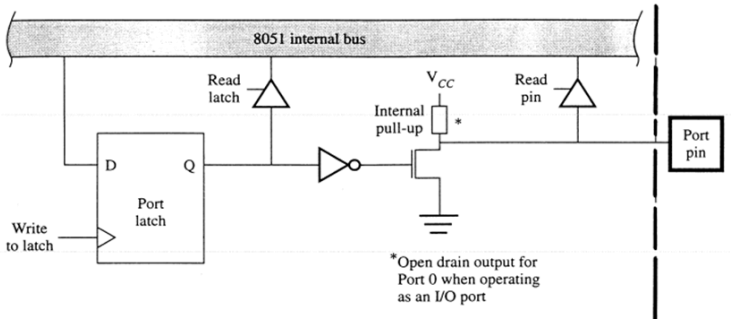
- **ALE (pin 30) : فعال ساز تثبیت آدرس**
- یک پین خروجی است و در وضعیت بالا فعال می شود و برای جداسازی گذرگاه داده و آدرس استفاده می شود.
  - درگاه ۰ در ۸۰۵۱ هم به عنوان گذرگاه داده و هم به عنوان بایت پایین گذرگاه آدرس به کار می رود.
  - از پین ALE برای دی مالتی پلکس کردن آدرس و داده توسط از اتصال به پایه G لچ **74LS373** استفاده می گردد.
- **پایه های درگاه ورودی/خروجی**
  - چهار درگاه P0، P1، P2 و P3 وجود دارد.
  - هر درگاه دارای ۸ پایه است.
  - تمامی پایه های ورودی خروجی دو جهته هستند.

- ۸۰۵۱ دارای چهار درگاه ورودی/خروجی است:
  - درگاه 0 (pins 32-39) : P0 (P0.0~P0.7)
    - در طراحی با کمترین اجزای ممکن به عنوان درگاه ورودی/خروجی همه منظوره به کار می رود.
    - در طراحی های بزرگتر به عنوان گذرگاه آدرس و داده مالتی پلکس شده استفاده می شود.
  - درگاه 1 (pins 1 - 8) : P1 (P1.0~P1.7)
    - در صورت نیاز برای ارتباط با وسایل خارجی به کار می روند.
  - درگاه 2 (pins 21-28) : P2 (P2.0~P2.7)
    - یک درگاه دو منظوره است.
    - به عنوان ورودی خروجی عمومی و یا بایت بالای آدرس در طراحی با حافظه خارجی به کار می رود
  - درگاه 3 (pins 10-17) : P3 (P3.0~P3.7)
    - یک درگاه دو منظوره است که غیر از ورودی خروجی عمومی هر یک از پایه های آن داری وظیفه خاص است.

بیت درگاه ۳	نام	پایه	عملکرد خاص
P3.0	RxD	10	دریافت داده برای درگاه سریال
P3.1	TxD	11	ارسال داده برای درگاه سریال
P3.2	$\overline{\text{INT0}}$	12	وقفه خارجی ۰
P3.3	$\overline{\text{INT1}}$	13	وقفه خارجی ۱
P3.4	T0	14	ورودی خارجی برای تایمر ۰
P3.5	T1	15	ورودی خارجی برای تایمر ۱
P3.6	$\overline{\text{WR}}$	16	سیگنال فعال ساز نوشتن در حافظه خارجی
P3.7	$\overline{\text{RD}}$	17	سیگنال فعال ساز خواندن از حافظه خارجی

ساختار درگاه I/O

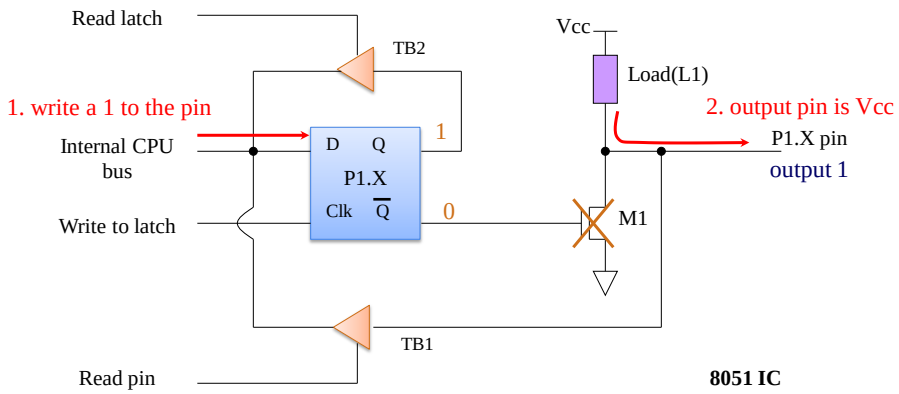
همانطور که گفته شد چهار درگاه ۸ بیتی **P0**، **P1**، **P2**و **P3**وجود دارد که همه آنها به استثنای**P1**که تنها برای **I/O**از آن استفاده می شود، دو منظوره هستند. شکل زیر دیاگرام یک بیت از درگاه **I/O**را در ۸۰۵۱ نشان می دهد.



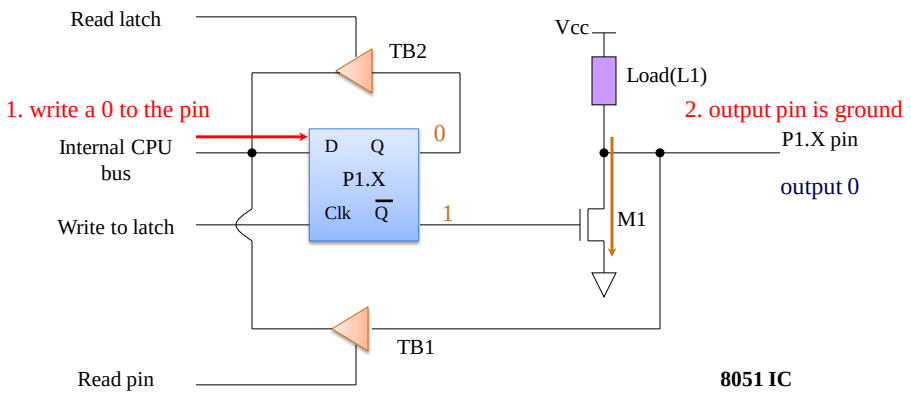
ساختار درگاه I/O

- هر پایه از درگاه های **I/O**:
  - گذرگاه **CPU** داخلی برای ارتباط با **CPU**
  - یک **D latch** برای ذخیره ساز ی مقدار پایه
    - **D latch** به وسیله **"Write to latch"** کنترل می گردد. اگر **Write to latch** = ۱ داده در **D latch** نوشته می شود.
  - ۲ بافر سه حالته
    - **TB1**: به وسیله **"Read pin"** کنترل می شود. اگر **Read pin=1** داده موجود در پایه خوانده می شود.
    - **TB2**: به وسیله **"Read latch"** کنترل می شود. اگر **Read latch=1** داده موجود در **latch** داخلی خوانده می شود.
  - گیت ترانزیستور
    - اگر **Gate=0**: باز
    - اگر **Gate=1**: بسته

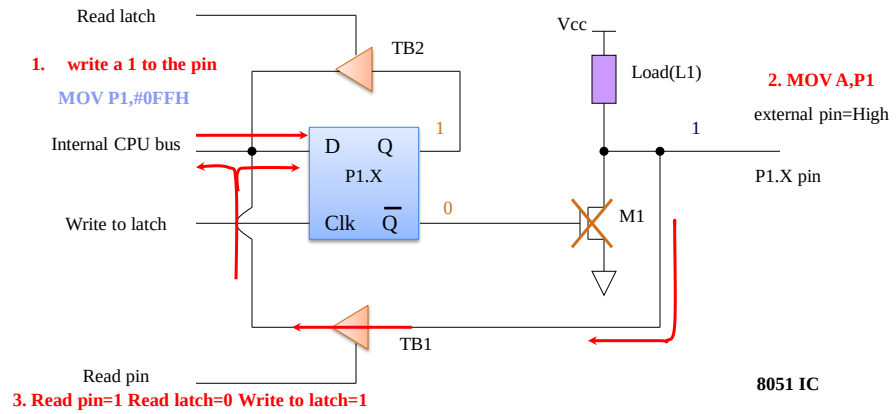
نوشتن «۱» در خروجی در پایه P1.X



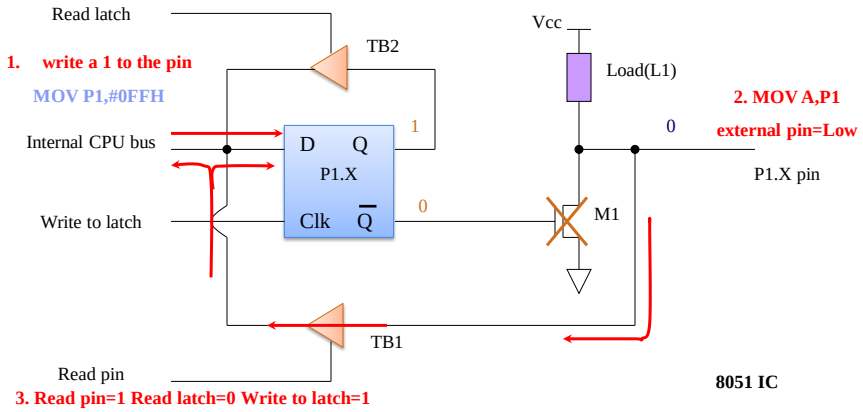
نوشتن «۰» در خروجی در پایه P1.X



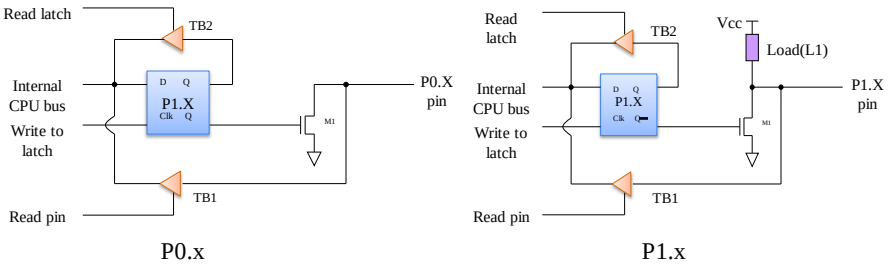
خواندن High در پایه ورودی



خواندن Lowدر پایه ورودی

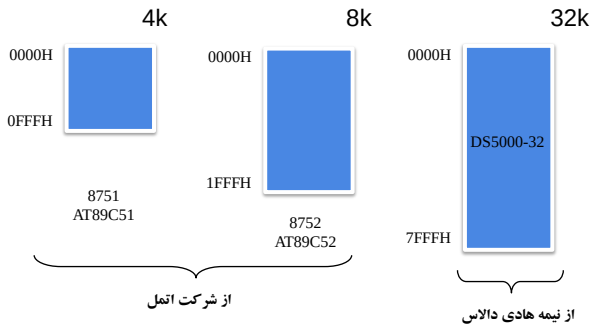


- P1, P2 و P3 دارای مقاومت های بالابر داخلی هستند.
  - P1, P2 و P3 درین باز نیستند.
- P0 دارای مقاومت های بالابر داخلی نمی باشند و در داخل ۸۰۵۱ به Vcc متصل نیستند.
  - P0 درین باز است.



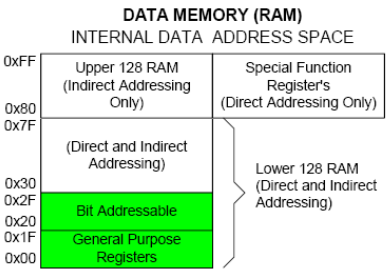
نقشه حافظه در ۸۰۸۵

نقشه حافظه ROM در خانواده ۸۰۵۱



نقشه حافظه در ۸۰۸۵

تخصیص فضای حافظه RAM در ۸۰۵۱



- فضای حافظه داخلی به سه دسته تقسیم می شود:
  - ۱۲۸ بایت بالایی
  - ۱۲۸ بایت پایینی
  - ثبات های کاربرد خاص **(SFR)**
- در اینجا به طور فیزیکی ۳۸۴ بایت حافظه وجود دارد، اگرچه ۱۲۸ بایت بالایی و **SFR**ها آدرس های همسانی از **80H** تا **FFH** را سهم می برند.
- باید دستورالعمل های مناسبی برای دستیابی به هر بلوک حافظه به کار رود.

نقشه حافظه در ۸۰۸۵

Byte Address	Bit Address							
7F	General Purpose RAM							
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Bank 3							
18								
17	Bank 2							
10								
0F	Bank 1							
08								
07	Default Register Bank for R0 – R7							
00								

Bit Address

بیت آدرس پذیر

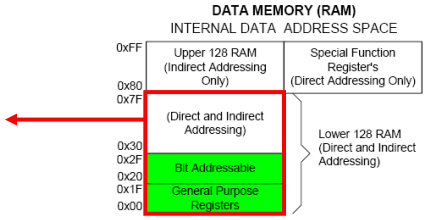
(۱۶ بایت)

RAM همه منظوره

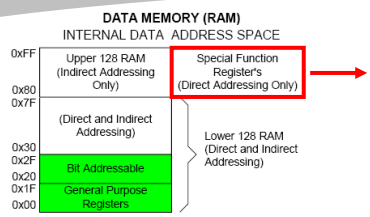
(۸۰ بایت)

بانک های ثبات

(۸ بایت برای هر بانک؛ ۴ بانک)



نقشه حافظه در ۸۰۸۵



- SFR ها امکان کنترل و تبادل اطلاعات با منابع میکروکنترلر و دستگاه های جانبی را فراهم می آورد.
- ثبات هایی که آدرس بایت آنها به 0H یا 8H ختم می گردد هم بیت و هم بایت آدرس پذیر هستند.
- برخی از ثبات ها بیت آدرس پذیر نیستند که شامل اشاره گر پشته SP و ثبات اشاره گر داده DPTR نیز می شود .

Byte Address	Bit Address								
FF									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	-	D0	PSW
B8	-	-	-	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	-	-	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	Not bit-addressable								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	Not bit-addressable								TH1
8C	Not bit-addressable								TH0
8B	Not bit-addressable								TL1
8A	Not bit-addressable								TL0
89	Not bit-addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	Not bit-addressable								PCON
83	Not bit-addressable								DPH
82	Not bit-addressable								DPL
81	Not bit-addressable								SP
80	87	86	85	84	83	82	81	80	P0

نماد	نام	آدرس
ACC*	Accumulator	0E0H
B*	B register	0F0H
PSW*	Program status word	0D0H
SP	Stack pointer	81H
DPTR	Data pointer 2 bytes	
DPL	Low byte	82H
DPH	High byte	83H
P0*	Port 0	80H
P1*	Port 1	90H
P2*	Port 2	0A0H
P3*	Port 3	0B0H
IP*	Interrupt priority control	0B8H
IE*	Interrupt enable control	0A8H
TMOD	Timer/counter mode control	89H
TCON*	Timer/counter control	88H
T2CON*	Timer/counter 2 control	0C8H
T2MOD	Timer/counter mode control	0C9H
TH0	Timer/counter 0 high byte	8CH
TL0	Timer/counter 0 low byte	8AH
TH1	Timer/counter 1 high byte	8DH
TL1	Timer/counter 1 low byte	8BH
TH2	Timer/counter 2 high byte	0CDH
TL2	Timer/counter 2 low byte	0CCH
RCAP2H	T/C 2 capture register high byte	0CBH
RCAP2L	T/C 2 capture register low byte	0CAH
SCON*	Serial control	98H
SBUF	Serial data buffer	99H
PCON	Power control	87H

* آدرس‌پذیر بیتی

لیستی از ثبات های کاربرد خاص و آدرس آنها

- **ثبات A یا ACC یا انباره**
  - انباره، همانطور که از نامش پیداست، یک ثبات همه منظوره برای انباشتن نتایج تعداد زیادی از دستورها می باشد . انباره می تواند یک مقدار ۸ بیتی (۱ بایتی) را در خود نگه دارد. بیش از نیمی از دستورات ۸۰۵۱ به نوعی با انباره سر و کار دارد.
- **ثبات B**
  - ثبات B نیز مانند A می تواند یک مقدار ۸ بیتی را در خود نگه دارد و به همراه انباره برای عملیات ضرب و تقسیم مورد استفاده قرار می گیرد. همچنین به عنوان یک ثبات موقتی (چرک نویس) همه منظوره عمل می نماید.

ثبات کلمه وضعیت برنامه PSW

ثبات کلمه وضعیت برنامه **PSW** همانطور که در جداول زیر نشان داده شده شامل بیت های وضعیت است.

CY	AC	F0	RS1	RS0	OV	--	P

بیت	نماد	ادرس	شرح عملکرد بیت
PSW.7	CY	D7H	پرچم نقلی
PSW.6	AC	D6H	پرچم نقلی کمکی
PSW.5	F0	D5H	پرچم 0
PSW.4	RS1	D4H	انتخاب بانک ثبات 0
PSW.3	RS0	D3H	انتخاب بانک ثبات 1
			00 = بانک 0 ; آدرس‌های 00H–07H
			01 = بانک 1 ; آدرس‌های 08H–0FH
			10 = بانک 2 ; آدرس‌های 10H–17H
			11 = بانک 3 ; آدرس‌های 18H–1FH
PSW.2	OV	D2H	پرچم سرریز
PSW.1	—	D1H	رزرو شده
PSW.0	P	D0H	پرچم توازن زوج

پرچم نقلی CY

- بیت نقلی یک بیت دو منظوره است:
  - زمانی که در یک عملیات محاسباتی، یک بیت نقلی از بیت ۷ خارج شود و یا یک بیت قرضی به بیت ۷ اضافه شود، پرچم نقلی یک می شود.
  - پرچم نقلی یک انباره بولی بوده که به عنوان یک ثبات تک بیتی در دستورالعمل های بولی عمل می کند.

پرچم نقلی کمکی AC

- چنانچه انتقالی از بیت ۳ به ۴ در جمع کردن اعداد کد شده به BCD رخ دهد، بیت نقلی کمکی یک خواهد شد.

پرچم 0

- بیت 0 یا F0 یک پرچم همه منظوره برای استفاده کاربران می باشد.

بیت های اتخاب بانک ثبات RS1 و RS0

- بیت های انتخاب بانک ثبات، بانک ثبات فعال را مشخص می کنند.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

پرچم سرریز OV

- چنانچه پس از یک عمل جمع یا تفریق سرریز حسابی روی دهد، این بیت ۱ می گردد. به عبارت دیگر نتایج بزرگتر از ۱۲۷+ یا کوچکتر از ۱۲۷- بیت OV را یک می نماید.

بیت توازن P

- بیت توازن در هر سیکل ماشین برای ایجاد توازن زوج انباره به طور خودکار ۰ یا ۱ می شود به این صورت که تعداد بیت های یک در انبارخ به علاوه بیت P همواره زوج است.

ثبات اشاره گر پشته SP

- ثباتی که برای دسترسی به پشته از آن استفاده می شود ثبات اشاره گر پشته sp نامیده می شود.
- اشاره گر پشته در ۸۰۵۱ دارای پهنای ۸ بیت می باشد و این بدان معنی است که مقادیری از 00 تا FFH را می تواند در خود جای دهد. زمانی که ۸۰۵۱ روشن می گردد، مقدار پیش فرض 07 می باشد.

ثبات اشاره گر داده DPTR

- اشاره گر داده یا DPTR برای دستیابی به حافظه کد یا داده خارجی مورد استفاده قرار می گیرد. این ثبات ۱۶ بیتی بوده و در آدرس های 82H (بایت پایین) و 83H (بایت بالا) قرار دارد.

ثبات های تایمر

- ۸۰۵۱ دارای دو تایمر ۱۶ بیتی T0 و T1 برای زمان بندی و شمارش اتفاقات است که عملکرد این تایمرها توسط ثبات حالت تایمر TMOD و ثبات کنترل تایمر TCON است.

ثبات های درگاه سریال

- ۸۰۵۱ دارای یک درگاه سریال بر روی تراشه است. ثباتی به نام بافر داده سریال SBUF داده های ارسالی و دریافتی را نگه می دارد. حالت های مختلف درگاه سریال نیز از طریق ثبات کنترل درگاه سریال SCON قابل برنامه ریزی می باشد.

ثبات وقفه

- ۸۰۵۱ دارای یک ساختار وقفه با ۵ منبع وقفه و ۲ سطح تقدم می باشد. وقفه ها با نوشتن در ثبات فعال سازی وقفه IE فعال می گردند و سطح تقدم از طریق ثبات تقدم وقفه IP تعیین می شود.

ثبات کنترل توان PCON

بیت	نماد	شرح عملکرد
7	SMOD	بیت دو برابر کننده نرخ ارسال؛ هنگام 1 شدن؛ نرخ ارسال در حالت‌های 1، 2 یا 3 درگاه سریال دو برابر می‌شود
6	—	تعریف نشده
5	—	تعریف نشده
4	—	تعریف نشده
3	GF1	بیت 1 پرچم همه منظوره
2	GF0	بیت 0 پرچم همه منظوره
1*	PD	افت تغذیه؛ برای فعال کردن حالت افت تغذیه 1 می‌شود تنها راه خروج، reset شدن است.
0*	IDL	حالت معلّق؛ برای فعال کردن حالت معلّق 1 می‌شود؛ تنها راه خروج، یکت وقفه یا reset شدن است.

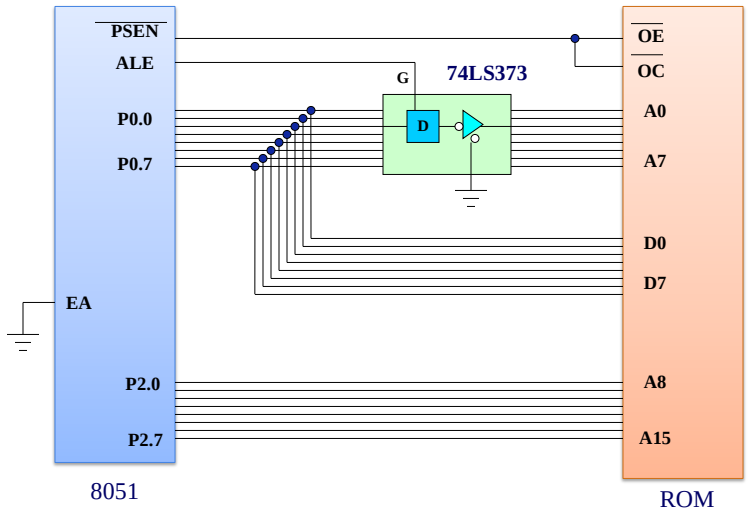
* تنها در نسخه‌های CMOS منظور شده است.

حافظه خارجی

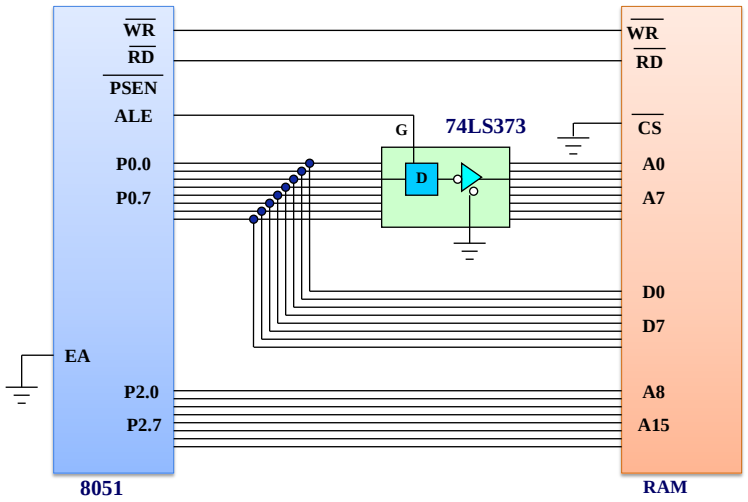
- زمانی که یک ۸۰۳۱/۸۰۵۱ را به یک حافظه خارجی وصل می کنیم، ۸۰۵۱ از پورت ها برای ارسال آدرس ها و خواندن دستورالعمل ها استفاده می کند.
  - ۸۰۳۱ قابلیت دسترسی به ۶۴ کیلو بایت حافظه خارجی را دارد.
  - ۱۶ بیت آدرس: P0 آدرس های A0-A7 و P2 آدرس های A8-A15 را فراهم می آورد.
  - همچنین P0 خط های داده D0-D7 را فراهم می سازد.

- زمانی که از P0 برای مالتی پلکس آدرس/داده استفاده می گردد، این درگاه برای نگه داشتن آدرس به 74LS373 متصل می گردد.
  - در این حالت نیازی به مقاومت های بالا بر نمی باشد.

دستیابی به حافظه کد خارجی

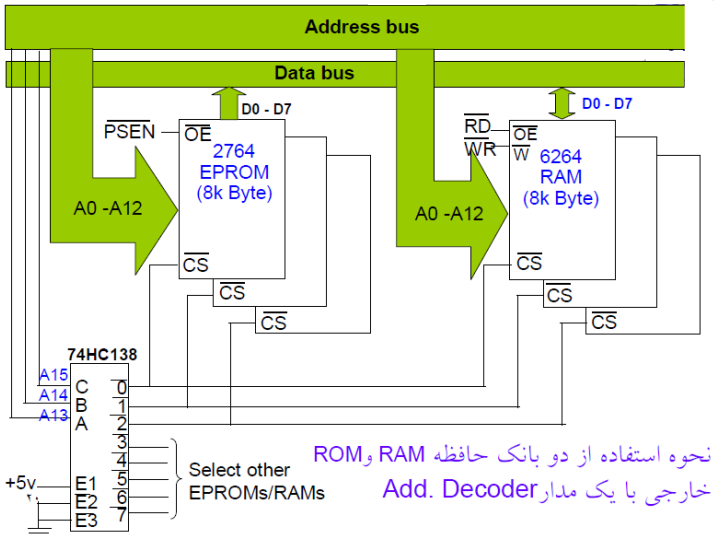


دستیابی به حافظه داده خارجی



۵۳

رمزگشایی آدرس



نحوه استفاده از دو بانک حافظه ROM و RAM
خارجی با یک مدار Add. Decoder

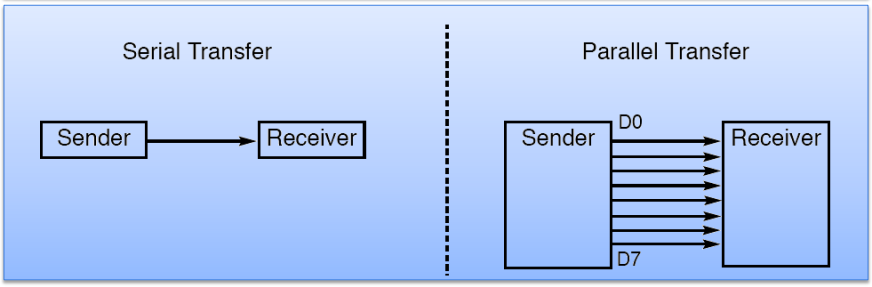
۵۴

فصل پنجم:

درگاه سریال

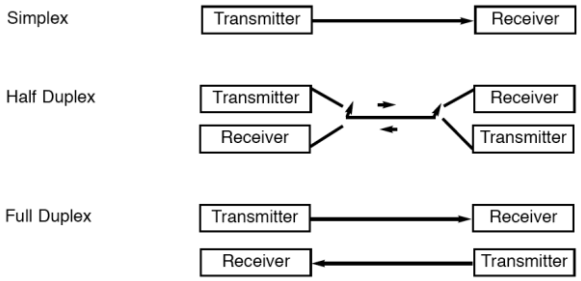
اصول ارتباط سریال

- **موازی:**
  - گران – سریع – فواصل کوتاه – بدون مدولاسیون
- **سریال:**
  - ارزان تر – کند تر – فواصل بلند (با استفاده از مودم)



اصول ارتباط سریال

- **دو روش سنکرون (همگام) و غیر سنکرون (ناهمگام)**
  - روش سنکرون بلوکی از داده (کارکترها) را در یک زمان منتقل می کند.
  - روش آسنکرون یک بایت را در هر زمان منتقل می کند.
- استفاده از تراشه هایی که آنها را UART (universal asynchronous receiver-transmitter) و USART (universal synchronous-asynchronous receiver-transmitter) می نامند



اصول ارتباط سریال

- **انتقال ساده *simplex*** کامپیوتر تنها اطلاعات را ارسال می کند.
- **انتقال دو طرفه کامل و نیمه *Half- and full-duplex***
  - چنانچه داده بتواند هم ارسال و هم دریافت گردد آنگاه انتقال دو طرفه خواهد بود.
  - انتقال دو طرفه می تواند بر اساس اینکه امکان ارسال و دریافت همزمان وجود داشته باشد یا نه، کامل یا نیمه باشد.
  - اگر تنها ارسال یا دریافت در یک زمان فعال باشد نیمه و اگر هر دو مورد در یک زمان فعال باشد انتقال دو طرفه کامل است.
  - انتقال دو طرفه کامل نیاز به دو سیم برای خطوط داده می باشد.

اصول ارتباط سریال

- **سرعت انتقال داده**
  - نرخ انتقال داده bps بیت در هر ثانیه
  - واژه مورد استفاده در بسیاری از موارد برای bps، نسبت علائم در ثانیه (باود، baud rate) است.
  - نرخ bps و باود لزوماً برابر نیست.
  - نرخ باود، به صورت تعداد تغییرات سیگنال در هر ثانیه تعریف می گردد.

اصول ارتباط سریال

- **ارتباط سریال ناهمگام و قاب بندی داده ها**
  - داده ها به صورت 0 و 1 انتقال می یابند.
  - برای درک بهتری از اطلاعات، فرستنده و گیرنده بر روی مجموعه ای از قوانین توافق می کنند.
  - پروتکل:

    داده ها چگونه بسته بندی می شوند.

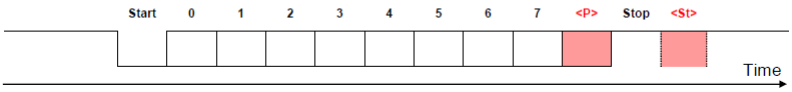
    چند بیت/کارکتر

    داده چه زمانی شروع و پایان می یابد.

اصول ارتباط سریال

- **بیت های شروع و پایان**
  - در روش ناهمگام، هر کارکتر بین بیت های شروع و پایان قرار می گیرند که به این قاب بندی گفته می شود.
  - بیت شروع معمولاً یک بیت و بیت پایان، یک یا دو بیت است.
  - بیت شروع همواره 0 و بیت پایان 1 است.
  - در آغاز LSB ارسال می گردد.

• Start bit (1→0), LSB, …, MSB, <parity bit>, Stop bit (0→1), <2nd stop bit (1)>

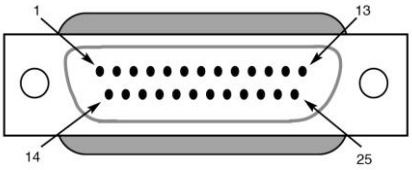


اصول ارتباط سریال

- **پروتکل RS232**
  - پرکاربردترین پروتکل واسط I/O سریال
  - سطوح ولتاژ ورودی و خروجی با TTL سازگار نیست.
  - بیت 1 با استفاده از 3- تا 25- ولت نشان داده می شود و بیت 0 از 3+ تا 25+ ولت است.
  - از 3- تا 3+ ولت تعریف نشده است.
  - برای اتصال RS232 به سک سیستم میکروکنترلری می بایست از یک مبدل ولتاژ مانند MAX232 برای تبدیل سطح منطقی TTL به سطوح ولتاژ RS232 و بالعکس استفاده نمود.

اصول ارتباط سریال

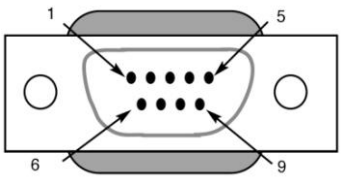
Pin	Description
1	Protective ground
2	Transmitted data (TxD)
3	Received data (RxD)
4	Request to send (RTS)
5	Clear to send (CTS)
6	Data set ready (DSR)
7	Signal ground (GND)
8	Data carrier detect (DCD)
9/10	Reserved for data testing
11	Unassigned
12	Secondary data carrier detect
13	Secondary clear to send
14	Secondary transmitted data
15	Transmit signal element timing
16	Secondary received data
17	Receive signal element timing
18	Unassigned
19	Secondary request to send
20	Data terminal ready (DTR)
21	Signal quality detector
22	Ring indicator
23	Data signal rate select
24	Transmit signal element timing
25	Unassigned



RS232 Connector DB-25

اصول ارتباط سریال

Pin	Description
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)



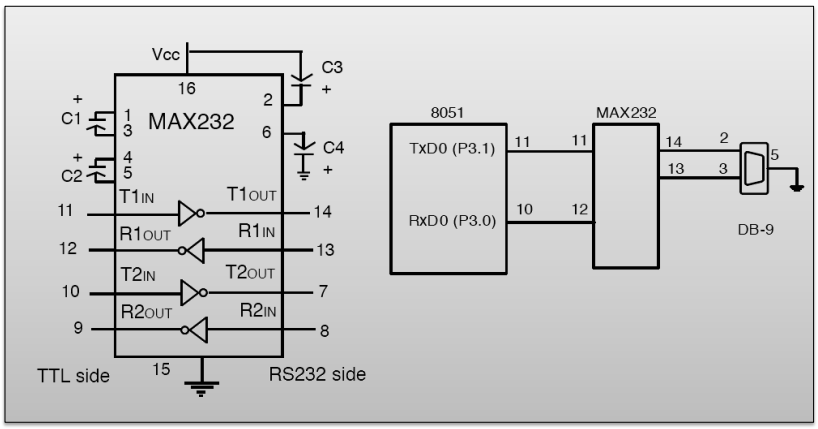
DB-9 9-Pin Connector

اصول ارتباط سریال

- **پروتکل RS232**
  - پرکاربردترین پروتکل واسط I/O سریال
  - سطوح ولتاژ ورودی و خروجی با TTL سازگار نیست.
  - بیت 1 با استفاده از 3- تا 25- ولت نشان داده می شود و بیت 0 از 3+ تا 25+ ولت است.
  - از 3- تا 3+ ولت تعریف نشده است.
  - برای اتصال RS232 به سک سیستم میکروکنترلری می بایست از یک مبدل ولتاژ مانند MAX232 برای تبدیل سطح منطقی TTL به سطوح ولتاژ RS232 و بالعکس استفاده نمود.

اصول ارتباط سریال

اتصال میکرو کنترلر 8051 به RS232



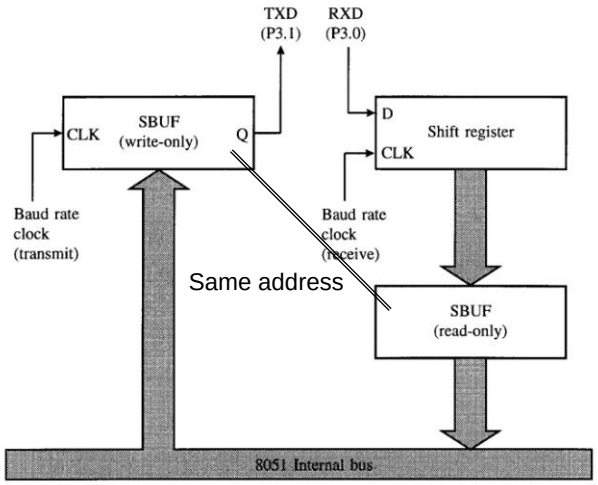
درگاه سریال در ۸۰۵۱

- ۸۰۵۱ یک درگاه سریال با عملکرد دوطرفه کامل دارد که می تواند به صورت یک واسط سریال نرمال (قاب بندی نشده) یا به عنوان یک UART (قاب بندی شده) مورد استفاده قرار بگیرد.
- این درگاه سریال از طریق RxD و TxD به صورت سخت افزاری در دسترس خواهد بود که عملکرد دیگر دو بیت از درگاه ۳ یعنی P3.0 (پایه ۱۰)، P3.1 (پایه ۱۱) می باشد.
- دو ثبات کاربرد خاص SBUF و SCON امکان دستیابی نرم افزاری را به درگاه سریال فراهم می آورند.

ثبات SBUF

- SBUF در واقع دو رجیستر مجزا در یک آدرس است:
  - ثبات ارسال فقط نوشتنی
  - ثبات دریافت فقط خواندنی
- نوشتن در SBUF داده ای را که باید ارسال شود بار می کند و خواندن SBUF داده دریافتی را در دسترس قرار می دهد.
- داده دریافتی تا زمانی که تمام بایت دریافت گردد در درگاه سریال بافر می گردد.
- داده قبلی می بایست تا قبل از اینکه بایت جدید کامل گردد خوانده شود.

ثبات SBUF



ثبات SCON

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM1،SM0:
مشخص کننده حالت عملکرد

SM0	SM1	حالت	عملکرد	نرخ انتقال
0	0	0	شیفت رجیستر	ثابت (xtal/12)
0	1	1	UART ۸بیت	متغیر (تایمر ۱)
1	0	2	UART ۹ بیت	ثابت(xtal/32 or xtal/64)
1	1	3	UART ۹ بیت	متغیر (تایمر ۱)

SM2: برای ارتباط چند پردازنده ای استفاده می گردد.

REN: فعال ساز گیرنده (فعال سازی/غیر فعال سازی به صورت نرم افزاری)

TB8: بیت ۸ ارسال

RB8: بیت ۸ دریافت

TI: پرچم وقفه ارسال. در پایان ارسال کارکتر ۱ می شود.

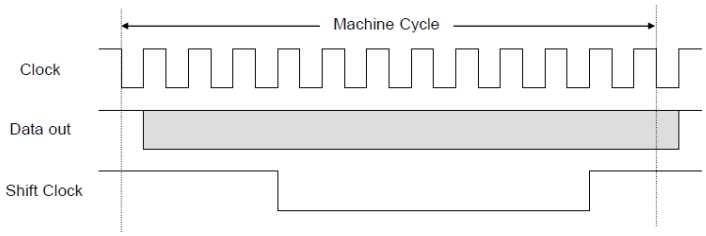
RI: پرچم وقفه دریافت. در پایان دریافت کارکتر ۱ می شود.

حالت های عملکرد درگاه سریال

- حالت 0، SM0 = SM1 = 0
  - عملکرد سنکرون نیمه دوطرفه
  - داده با استفاده از پایه RxD ارسال و دریافت می گردد (نه به صورت همزمان)
  - پایه TxD ساعت انتقال را هم در ارسال و هم در دریافت حمل می کند.
  - داده ها به صورت ۸ بیتی در بسته های بدون قاب بندی ارسال می گردند.
  - ابتدا کم ارزش ترین بیت انتقال می یابد.
  - نرخ انتقال در 1/12 فرکانس نوسان ساز روی تراشه ثابت است.

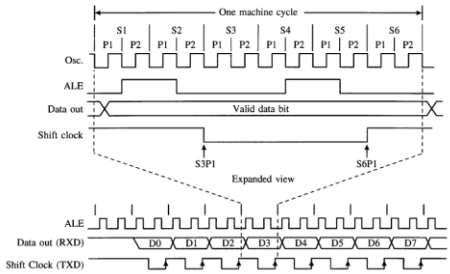
حالت ۰- ارسال

- انتقال به محض اینکه بایت در SBUF نوشته می شود آغاز می گردد.
- در هنگام انتقال، هر بیت در پایه RxD برای یک سیکل ماشین کامل معتبر باقی می ماند.
  - ساعت انتقال در وسط سیکل پایین می رود و دوباره قبل از تمام شدن سیکل بالا بر می گردد.
- پرچم وقفه ارسال زمانی که هشتمین بیت انتقال یافت یک می گردد.



حالت ۰- دریافت

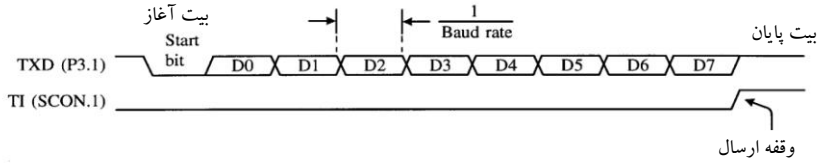
- انتقال به محض اینکه بیت REN یک گردد و پرچم وقفه دریافت پاک گردد، آغاز می گردد.
- معمولاً REN در آغاز برنامه برای مقدار دهی اولیه درگاه سریال یک می گردد سپس پرچم وقفه دریافت برای آغاز دریافت پاک می گردد.
- به محض اینکه پرچم وقفه دریافت پاک گردید، ساعت انتقال در پایه TxD تولید می گردد.
- کلاکینگ در زمان لبه بالارونده خط TxD اتفاق می افتد. پس از هشتمین سیکل کلاکینگ، داده در SBUF کپی گردیده و پرچم وقفه دریافت ۱ می گردد.

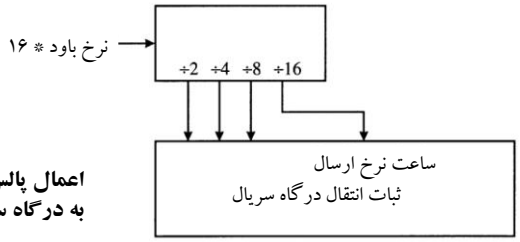


حالت ۱- انتقال

- در حالت ۱ درگاه سریال ۸۰۵۱ به عنوان یک UART ۸بیتی با نرخ باود متغیر عمل میکند .
- عملکرد اصلی یک UART تبدیل موازی به سریال داده خروجی و تبدیل سریال به موازی داده ورودی است.
- در حالت ۱، ۱۰ بیت روی TxD ارسال یا روی RxD دریافت می گردد
  - ۱۰ بیت: ۱ بیت آغاز (همیشه صفر)، ۸ بیت داده، ۱ بیت پایان (همیشه یک)
- نرخ ارسال بیت توسط نرخ سرریز تایمر ۱ تعیین می شود.
- عمل ارسال با نوشتن در SBUF آغاز می گردد.
- پرچم وقفه ارسال به محض اینکه بیت پایان روی خط TxD پدیدار گردد یک می شود.

حالت ۱ – انتقال





اعمال پالس های ساعت به درگاه سریال

حالت ۱ – دریافت

- دریافت با انتقال از ۱ به ۰ در خط RxD (با فرض اینکه REN ۱ است) آغاز می گردد.
  - شمارنده تقسیم بر ۱۶ فوراً پاک می شود تا بیت های ورودی را ردیف کند. بیت بعدی در چرخه بعدی تقسیم بر ۱۶ می رسد و به همین ترتیب ادامه می یابد.
  - بیت ها در هشتمین شماره از این شمارنده نمونه برداری می شود.
- تشخیص خطای بیت آغاز:
  - هشت شمارش پس از انتقال ۱ به ۰ خط RxD دوباره نمونه برداری می شود چنانچه صفر نباشد خطا در بیت شروع داریم. دریافت کننده راه اندازی مجدد شده و تا گذر بعدی از ۱ به ۰ منتظر می گردد.

حالت ۱ - دریافت

- با فرض تشخیص یک بیت آغاز معتبر، دریافت کارکتر ادامه می یابد. پس از انتقال هر هشت بیت:
  - بیت نهم (بیت پایان) به RB8 در Scon منتقل می گردد.
  - SBUF توسط ۸ بیت داده بارگذاری می گردد.
  - پرچم وقفه دریافت یک می گردد.
- اعمال فوق هنگامی اتفاق می افتد که شرایط زیر برقرار باشد:
  - RI=0
  - SM2=1 و بیت پایان دریافتی برابر 1 باشد و یا اینکه SM2=0 باشد.

حالت ۲

- درگاه سریال به عنوان یک UART ۹ بیتی با نرخ باود ثابت عمل می نماید.
- ۱۱ بیت ارسال می گردد:
  - بیت آغاز
  - ۸بیت داده از SBUF
  - نهمین بیت داده از TB8
  - بیت پایان
- در دریافت بیت نهم داده در RB8 قرار داده می شود.
- نرخ انتقال در 1/32 یا 1/64 فرکانس نوسانساز ثابت است.

حالت ۳

- درگاه سریال به عنوان یک UART ۹ بیتی با نرخ ارسال متغیر عمل می نماید.
- نرخ ارسال قابل برنامه ریزی است و توسط تایمر فراهم می گردد.
- ترکیبی از حالت ۱ و ۲ می باشد.

نرخ انتقال

- در حالت ۰ نرخ انتقال در فرکانس نوسان ساز تقسیم بر ۱۲ ثابت است.
- به صورت پیش فرض نرخ انتقال در حالت ۲ در 1/64 فرکانس نوسان ساز تنظیم می گردد.
  - اگر بیت ۷ از ثبات PCON (کنترل توان) – به عنوان SMOD شناخته می شود- یک گردد نرخ انتقال دو برابر می گردد.
  - اگر SMOD=1 باشد نرخ انتقال در حالت دو 1/32 فرکانس نوسان ساز است.
- در حالت ۱ و ۳ نرخ انتقال با استفاده از نرخ سرریز تایمر ۱ تعیین می گردد.
  - چون این نرخ بسیار بالا است، به ۳۲ (یا به ۱۶ چنانچه SMOD=1) برای تولید نرخ واقعی تقسیم می گردد

تنظیم تایمر ۱ به منظور تولید نرخ انتقال

- به عنوان مثال، چگونه می توانیم نرخ انتقال ۱۲۰۰ را با استفاده از فرکانس نوسان ساز ۱۲ MHz تولید نماییم؟

- نرخ انتقال = K* FREC/(32*12*[256-TH1])
- برای تولید نرخ انتقال ۱۲۰۰ می بایست تایمر ۱ را طوری تنظیم نماییم که ۲۳ شماره را بشمارد.
- تایمر ۱ را طوری تنظیم می کنیم که در حالت ۲ (بارگذاری خودکار) عمل کند و TH1 را 0E6H(-23) قرار می دهیم.

۱۲۷

مراحل ارسال یک بایت

- برنامه ریزی T1 برای حالت ۲
- بارگذاری TH1 با مقدار اولیه
- برنامه ریزی SCON برای حالت ۱
- آغاز به کار تایمر ۱
- پاک کردن TI
- بارگذاری SBUF
- انتظار تا TI یک گردد.
- رفتن به مرحله پنجم برای بایت بعدی

۱۲۸

مثال: ارسال یک کارکتر

- برنامه ای برای انتقال ASCII "A" در نرخ ۹۶۰۰ به طور سریال

```
START:   MOV TMOD, #20H      ;Put T1 in mode2
         MOV TH1, #-3        ;9600 baud
         MOV SCON, #50H      ;8b, 1stop, 1start, REN enabled
         SETB TR1            ;start timer T1
AGAIN:   CLR TI              ;ready to transmit
         MOV SBUF, #'A'      ;letter A is to be transmitted
HERE:    JNB TI, HERE        ;poll TI until all the bits are transmitted
         SJMP AGAIN          ;while(1) loop (forever loop)
```


مراحل دریافت یک بایت

- برنامه ریزی T1 برای حالت ۲
- بارگذاری TH1 با مقدار اولیه
- برنامه ریزی SCON برای حالت ۱
- آغاز به کار تایمر ۱
- پاک کردن RI
- انتظار تا RI یک گردد.
- ذخیره SBUF
- رفتن به مرحله پنجم برای بایت بعدی

مثال: دریافت یک کارکتر

- برنامه ای برای دریافت بایت به طور سریال و نمایش آن در P1 به طور مداوم

```
START:   MOV TMOD, #20H    ;T1 in mode 2
         MOV TH1, #-3      ;9600 baud
         MOV SCON, #50H    ;8b, 1start, 1stop
         SETB TR1          ;start T1
AGAIN:   CLR RI            ;ready to receive a byte
HERE:    JNB RI, HERE      ;wait until one byte is Rx-ed
         MOV A, SBUF       ;read the received byte from SBUF
         MOV P1, A         ;display on P1
         SJMP AGAIN        ;while (1)
```

فصل چهارم:

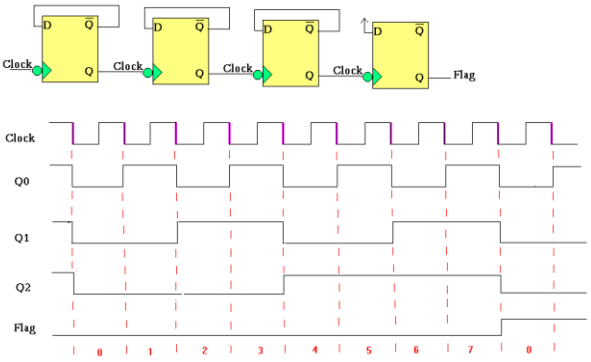
تایمر

تایمرها در ۸۰۵۱

- ۸۰۵۱ دو تایمر دارد:
  - تایمر ۰
  - تایمر ۱
- کاربردهای تایمر:
  - تولید کننده تاخیر زمانی
    - منبع ساعت، فرکانس داخلی کریستال ۸۰۵۱ می باشد
  - شمارش اتفاقات
    - ورودی خارجی برای ورودی پین جهت شمارش اتفاقات
    - پالس های یاعت می تواند نمایشگر تعداد افرادی که از یک ورودی می گذرند باشد یا تعداد چرخش یک چرخ یا هر رخداد دیگری که می تواند تبدیل به پالس شود می باشد.
  - تولید نرخ ارسال بیت بیت برای درگاه سریال داخلی

تایمرها در ۸۰۵۱

- تایمر از چند فلیپ فلاپ تقسیم کننده بر دو تشکیل می شود.
- یک تایمر با n فلیپ فلاپ، فرکانس ساعت ورودی را بر 2^n تقسیم می کند.
- خروجی آخرین فلیپ فلاپ ورودی ساعت یک فلیپ فلاپ به نام پرچم یا سرریز خواهد بود.



تایمرها در ۸۰۵۱

- تایمرهای ۸۰۵۱ با استفاده از شش ثبات کاربرد خاص قابل دستیابی هستند.
- ۵ ثبات کاربرد خاص اضافه تر برای دسترسی به تایمر سوم در ۸۰۵۲ وجود دارد.

ثبات‌های کاربرد خاص تایمر

SFR تایمر	کاربرد	آدرس	بیت آدرس‌پذیر
TCON	کنترل	88H	بله
TMOD	حالت	89H	نه
TL0	بایت پایین تایمر 0	8AH	نه
TL1	بایت پایین تایمر 1	8BH	نه
TH0	بایت بالای تایمر 0	8CH	نه
TH1	بایت بالای تایمر 1	8DH	نه
T2CON*	کنترل تایمر 2	C8H	بله
RCAP2L*	بایت پایین تسخیری در تایمر 2	CAH	نه
RCAP2H*	بایت بالای تسخیری در تایمر 2	CBH	نه
TL2*	بایت پایین تایمر 2	CCH	نه
TH2*	بایت بالای تایمر 2	CDH	نه

ثبات TMOD

- این ثبات دارای دو گروه چهار بیتی برای تعیین حالت کار تایمر ۰ و تایمر ۱ می باشد.
- این ثبات بیت آدرس پذیر نبوده و معمولاً یک بار به صورت نرم افزاری در ابتدای برنامه برای مقدار دهی اولیه به حالت کار تایمر بارگذاری می شود.

بیت	نام	تایمر	شرح عملکرد
7	GATE	1	بیت GATE. هنگام 1 بودن تایمر، فقط وقتی به راه می‌افتد که $\overline{INT1}$، 1 باشد
6	C/ $\overline{T}$	1	بیت انتخاب تایمر/شمارنده شمارش اتفاقات = 1 زمان‌بندی فاصله‌های زمانی = 0
5	M1	1	(جدول ۳-۴) بیت حالت 1
4	M0	1	(جدول ۳-۴) بیت حالت 0
3	GATE	0	بیت گیت تایمر 0
2	C/ $\overline{T}$	0	بیت انتخاب تایمر/شمارنده برای تایمر 0
1	M1	0	بیت M1 تایمر 0
0	M0	0	بیت M0 تایمر 0

Gate	C/$\overline{T}$	M1	M0	Gate	C/$\overline{T}$	M1	M0
Timer1				Timer0			

M1	M0	حالت	شرح عملکرد
0	0	0	حالت تایمر ۱۳ بیتی (حالت 8048)
0	1	1	حالت تایمر ۱۶ بیتی
1	0	2	حالت بارشدن خودکار، ۸ بیتی
1	1	3	حالت تایمر دو قسمتی: تایمر 0 : TL0 یک تایمر ۸ بیتی است که توسط بیت‌های حالت تایمر 0 کنترل می‌شود؛ TH0 هم همان‌طور فقط با بیت‌های حالت تایمر 1 کنترل می‌شود. تایمر 1: موقف است

ثبات TMOD

- GATE=0
  - کنترل داخلی
  - شروع و توقف تایمر به صورت نرم افزاری صورت می گیرد.
  - **set/clear** کردن **TR** برای شروع/توقف تایمر

```
SETB   TR0
CLR    TR0
```

- GATE=1
  - کنترل خارجی
  - روش سخت افزاری آغاز به کار و توقف تایمر با استفاده از نرم افزار و یک منبع خارجی
  - تایمر/شمارنده تنها زمانی فعال می شوند که پایه **INT** بالا شود و پین کنترل **TR** یک گردد.

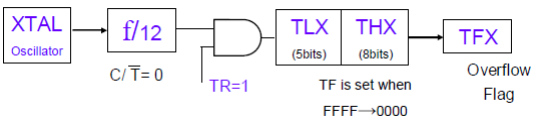
ثبات TCON

بیت	نماد	آدرس بیت	شرح عملکرد
TCON.7	TF1	8FH	پرچم سرریز تایمر 1: بصورت سخت‌افزاری هنگام سرریز 1 می‌شود. بصورت نرم‌افزاری پاک می‌شود و یا بصورت سخت‌افزاری هنگامی که پردازنده روال سرویس وقفه را نشانه رفته است.
TCON.6	TR1	8EH	بیت کنترل روندشمارش تایمر 1 بصورت نرم‌افزاری برای روشن - خاموش کردن تایمر، 1 یا 0 می‌شود.
TCON.5	TF0	8DH	پرچم سرریز تایمر 0
TCON.4	TR0	8CH	بیت کنترل روندشمارش تایمر 0
TCON.3	IE1	8BH	پرچم لبه وقفه خارجی 1. بصورت سخت‌افزاری وقتی که لبه پایین رونده در INT1 تشخیص داده شود 1 می‌گردد؛ بصورت نرم‌افزاری یا توسط سخت‌افزار وقتی که CPU روال سرویس وقفه را نشانه رفته است پاک می‌شود.
TCON.2	IT1	8AH	پرچم نوع وقفه خارجی 1. بصورت نرم افزاری برای وقفه‌های خارجی فعال شونده با سطح منطقی صفر/لبه پایین رونده 0/1 می‌شود.
TCON.1	IE0	89H	پرچم لبه وقفه خارجی 0
TCON.0	IT0	88H	پرچم نوع وقفه خارجی 0

- چهار بیت بالایی (۷-۴) برای روشن و خاموش کردن تایمرها **(TR0,TR1)** یا پیام سرریز کردن تایمر **(TF0,TF1)** به کار می رود.
- چهار بیت پایینی کاری برای تایمر انجام نخواهد داد. بلکه برای تشخیص و مقدار دهی اولیه وقفه ها به کار می رود.

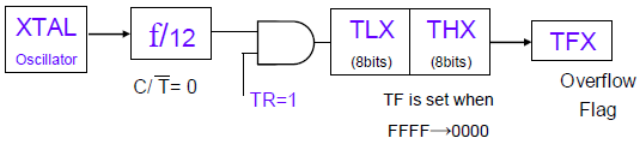
حالت تایمر ۱۳ بیتی (حالت ۰)

- حالت ۰ یک حالت ۱۳ بیتی برای تایمر است.
- این حالت نشان دهنده سازگاری این میکرو با پردازنده قبل از خود ۸۰۴۸ می باشد.
- در طراحی های جدید معمولاً از این حالت استفاده نمی شود.
- تایمر ۱۳ بیتی از به هم پیوستن بایت بالای تایمر (THx) و ۵ بیت از بایت پایین تایمر (TLx) تشکیل شده.
- از سه بیت بالایی بایت پایین تایمر (TLx) در این حالت استفاده نمی شود.



حالت تایمر ۱۶ بیتی (حالت ۱)

- حالت ۱۶بیتی مشابه حالت ۰ بوده با این تفاوت که تایمر به صورت یک تایمر ۱۶ بیتی کامل کار می کند.
- با دریافت پالس ساعت تایمر از 0000 تا FFFF شمارش می کند. در گذر از FFFF به 0000 سر ریز اتفاق می افتد و پرچم سرریز، بیت TFx در TCON یک می گردد.
- پر ارزش ترین بیت تایمر در این حالت بیت ۷ در THx و کم ارزش ترین بیت، بیت ۰ در TLx می باشد.



مراحل حالت ۱

۱. انتخاب تایمر ۰ و حالت ۱

- MOV TMOD,#01H

۲. تعیین مقادیر اولیه برای TL0 و TH0

- MOV TH0,#FFH
- MOV TL0,#FCH

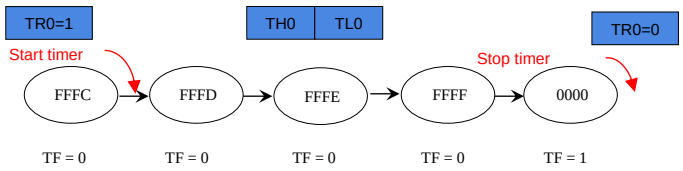
۳. بهتر است که پرچم را پاک نمایید. TF0=0

- CLR TF0

۴. تایمر را روشن کنید

- SETB TR0

۵. ۸۰۵۱ با افزایش TL0 و TH0 شروع به شمارش به سمت بالا می کند.



مراحل حالت ۱

۶. زمانی که TL0 – TH0 از FFFFH به 0000 سرریز می کند ۸۰۵۱، TF را یک می کند.

۷. پایش پرچم تایمر را دنبال کنید تا متوجه شوید افزایش یافته

- AGAIN: JNB TF0, AGAIN

۸. TR0 را برای متوقف کردن فرآیند پاک کنید.

- CLR TR0

۹. پرچم TF را برای دور بعدی پاک نمایید.

- CLR TF0

محاسبه تاخیر تایمر برای XTAL = 11.0592 MHz

الف) هگز

- $(FFFF - YYXX + 1) \times 1.085\ \mu s$
- در اینجا YYXX، به ترتیب مقادیر اولیه TH, TL هستند.
- توجه داشته باشید که مقدار YYXX یک مقدار هگز است.

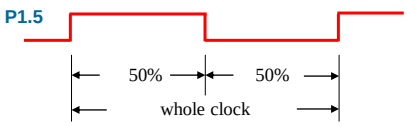
ب) ده دهی

- مقدار YYXX به دست آمده از ثبات های TH, TL را به ده دهی تبدیل نمایید تا به عدد ده دهی NNNNN برسید.
- سپس $(65536 - NNNNN) \times 1.085\ \mu s$

مثال

- با استفاده از تایمر ۰ یک موج مربعی با چرخه کاری 50% بر روی پایه P1.5 ایجاد نمایید.

```
        MOV TMOD,#01          ;Timer 0,mode 1(16-bit)
HERE:   MOV TL0,#0F2H         ;Timer value = FFF2H
        MOV TH0,#0FFH
        CPL P1.5
        ACALL   DELAY
        SJMP    HERE
DELAY:
        SETB TR0              ;start the timer 0
AGAIN:  JNB TF0,AGAIN
        CLR TR0               ;stop timer 0
        CLR TF0               ;clear timer 0 flag
        RET
```



- **0000 0001** = TMOD قرار داده شد.
- **FFF2H** در TL0 – TH0 بارگذاری شد.
- P1.5 برای پایین و بالا رفتن پالس کنترل می شود.
- زیر روال DELAY که از تایمر استفاده می کند فراخوانی می شود.
- در زیرروال DELAY، تایمر ۰ با دستور **SETB TR0** آغاز به کار می کند.
- تایمر ۰ با گذشتن هر کلاک تولید شده به وسیله کریستال نوسان ساز شمارش رو به بالا را انجام می دهد:

FFF3, FFF4, FFF5, FFF6, FFF7, FFF8, FFF9, FFFA, FFFB, FFFC, FFFFD, FFFE, FFFFH یک کلاک دیگر آن را ۰ می کند و پرچم تایمر یک می شود. در این نقطه دستور JNB می شکند.

- تایمر ۰ با دستور **CLR TR0** متوقف می گردد. زیر روال DELAY به پایان می رسد و فرآیند تکرار می شود.

مثال

- این برنامه یک موج مربعی بر روی P1.5 با استفاده از تایمر ۱ ایجاد می کند. با فرض XTAL = 11.0592 MHz فرکانس را بیابید.

```
        MOV  TMOD,#10H          ;timer 1, mode 1
AGAIN:  MOV  TL1,#34H           ;timer value=3476H
        MOV  TH1,#76H
        SETB TR1                ;start
BACK:   JNB  TF1,BACK
        CLR  TR1                ;stop
        CPL  P1.5               ;next half clock
        CLR  TF1                ;clear timer flag 1
        SJMP AGAIN
```

Solution:
FFFFH – 7634H + 1 = 89CCH = 35276 clock count
Half period = 35276 × 1.085 µs = 38.274 ms
Whole period = 2 × 38.274 ms = 76.548 ms
Frequency = 1/ 76.548 ms = 13.064 Hz.

محاسبه مقدار تایمر

- فرض کنید که XTAL = 11.0592 MHz و ما مقدار تاخیر مورد نظر را می دانیم. چگونه می توان مقادیر TH,TL را به دست آورد؟
  - تاخیر را بر 1.085 s تقسیم می کنیم تا n به دست آید.
  - n را از ۶۵۵۳۶ کم کنید.
  - نتیجه به دست آمده را به مبنای ۱۶ تبدیل کنید. (yyxx)
  - TH = yy و TL = xx قرار دهید.

مثال

- با فرض XTAL = 11.0592 MHz برنامه ای بنویسید که یک موج مربعی با فرکانس 50 Hz بر روی P2.3 تولید نماید.

- حل:

دوره تناوب موج مربعی برابر است با : 1 / 50 Hz = 20 ms

بخش بالا یا پایین موج مربعی برابر است با: 10 ms

10 ms / 1.085 us = 9216

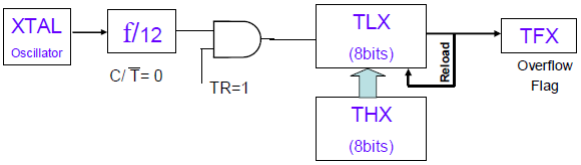
65536 – 9216 = 56320 in decimal = DC00H in hex.

TH1 = DCH و TL1 = 00H


```
        MOV  TMOD,#10H          ;timer 1, mode 1
AGAIN:  MOV  TL1,#00            ;Timer value = DC00H
        MOV  TH1,#0DCH
        SETB TR1                ;start
BACK:   JNB  TF1,BACK
        CLR  TR1                ;stop
        CPL  P2.3
        CLR  TF1                ;clear timer flag 1
        SJMP AGAIN              ;reload timer since  mode 1 is not
                                ;auto-reload
```

حالت بار شدن خودکار ۸ بیتی (حالت ۲)

- این حالت یک حالت ۸ بیتی با بارگیری مجدد تایمر از خودش می باشد.
- بایت پایین تایمر (TLx) به عنوان یک تایمر ۸ بیتی عمل می کند و بایت بالای تایمر (THx) مقداری را برای بارگیری مجدد در خود نگهداری می کند.
- زمانی که شمارنده از FF به 00 سرریز می کند مقدار THx به TLx بارگذاری می شود.



مراحل حالت ۲

۱. انتخاب تایمر ۰ و حالت ۲

- MOV TMOD,#02H

۲. تعیین مقدار اولیه برای TH0

- MOV TH0,#38H

۳. پرچم را پاک نمایید. TF0=0

- CLR TF0

۴. پس از اینکه TH0 با یک مقدار ۸ بیتی بارگذاری شد، ۸۰۵۱ یک کپی از آن در TL0 می گیرد.

- TL0=TH0=38H

۵. آغاز به کار تایمر

- SETB TR0

۶. ۸۰۵۱ با افزایش TL0 شمارش را انجام می دهد.

- TL0= 38H, 39H, 3AH,....

مراحل حالت ۲

۷. زمانی که TL0 از FFH به 00H سر ریز می کند ۸۰۵۱ TF0 را یک می کند. همچنین TL0 به طور خودکار با مقدار نگهداری شده در TH0 بارگذاری می گردد.

- TL0= FEH, FFH, 00H (Now TF0=1)
- بارگذاری خودکار 8051
- TL0=TH0=38H.
- Clr TF0
- رفتن به مرحله ۶

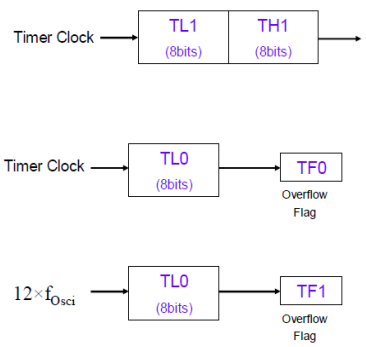
توجه داشته باشید که می بایست زمانی که TL0 سرریز می گردد TF0 را پاک نماییم. به این طریق می توانیم TF0 را در فرآیندهای بعدی زیر نظر بگیریم.

برای متوقف کردن فرآیند TR0 را پاک نمایید.

- Clr TR0

حالت تایمر تفکیک شده (حالت ۳)

- حالت ۳ برای هر تایمر متفاوت می باشد.
- تایمر ۰ در حالت ۳ به صورت دو تایمر ۸ بیتی مجزا (TL0 و TH0) عمل کرده و بیت های سر ریز آنها TF0 و TF1 می باشد.
- تایمر ۱ در حالت ۳ متوقف است.
- تایمر ۱ می تواند به حالت های دیگر سوییچ کند و برای تولید نرخ ارسال بیت به درگاه سریال و یا هر کاربرد دیگری که به وقفه نیاز ندارد مورد استفاده قرار بگیرد چراکه به TF1 وصل نیست.



شمارنده

- همانطوری که گفته شد از تایمرها به عنوان شمارنده رخدادهایی که در خارج از ۸۰۵۱ اتفاق می افتد می توان استفاده کرد.
- زمانی که از یک تایمر به عنوان شمارنده استفاده می شود، پالسی در خارج از ۸۰۵۱ وجود دارد که موجب بالا رفتن TH, TL می گردد.
- زمانی که C/T=1 می شود، از طریق پالس های تغذیه شده از:
  - T0: ورودی تایمر ۰ (پایه ۱۴، P3.4)
  - T1: ورودی تایمر ۱ (پایه ۱۵، P3.5)

  به سمت بالا شمارش می کند.

انتخاب تایمر/شمارنده در ۸۰۵۱

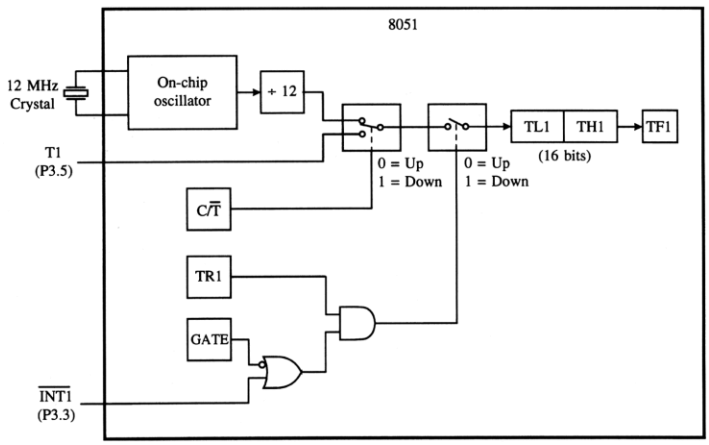
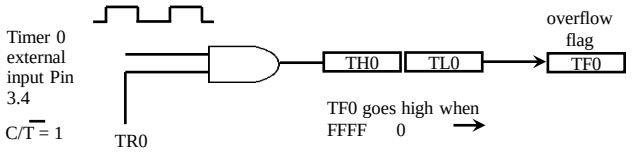


FIGURE 4–5
Timer 1 operating in mode 1

حالت ۱ شمارنده

- حالت ۱ شمارنده ۱۶ بیتی است. TH0 و TL0
- TH0 و TL0 زمانی که TR0 یک می گردد و پالس خارجی در T0 اعمال می گردد شروع به افزایش می کند.
- زمانی که شمارنده به حداکثر مقدار خود یعنی FFFFH می رسد، به مقدار 0000H سرریز می کند و TF0 یک می گردد.
- برنامه نویس می تواند به TH0 و TL0 مقدار اولیه تخصیص دهد و TF0=1 نماید تا از یک اندیکاتور برای نمایش شرایط خاصی استفاده نماید. (به عنوان مثال ۱۰۰ نفر آمده اند)



حالت ۲ شمارنده

- حالت ۲ شمارنده ۸ بیتی است.
  - در TH0 تنها مقادیری از 00 تا FF بارگذاری می گردد.
- بارگذاری خودکار TL0 چنانچه TR0 برابر یک گردد و پالس خروجی اتفاق بیافتد، روی می دهد.

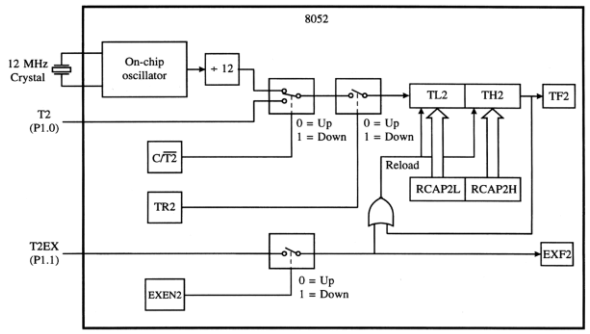


FIGURE 4–8
Timer 2 in 16-bit auto-reload mode

حالت ۲ شمارنده

- فرض کنید که پالس های ساعت پین T1 را تغذیه می کند. برنامه ای بنویسید که پالس ها را شمارش نماید و وضعیت TL1 و شمارش را در P2 نشان دهد.
- حل:

```
        MOV     TMOD,#01100000B     ;mode 2, counter 1
        MOV     TH1,#0
        SETB    P3.5                ;make T1 input port
AGAIN   :SETB   TR1                 ;start
BACK:   MOV     A,TL1
        MOV     P2,A                ;display in P2
        JNB     TF1,Back            ;overflow
        CLR     TR1                 ;stop
        CLR     TF1                 ;make TF=0
        SJMP    AGAIN                ;keep doing it
```

فصل سوم:

خلاصه مجموعه دستورالعمل ها

دستورالعمل ها

- دستورالعمل های ۸۰۵۱ مانند هر پردازنده ۸ بیتی، کدهایی ۸ بیتی دارند.
- کدهای ۸ بیتی امکان ۲۵۶=$۲^۸$ دستورالعمل را فراهم می سازد.
- از ۲۵۶ دستورالعمل، تعداد ۲۵۵ دستورالعمل قابل اجرا و یک دستورالعمل تعریف نشده است.
- برخی از دستورالعمل ها علاوه بر کد عملیاتی، یک یا دو بایت اضافی برای داده یا آدرس دارند.
- ۸۰۵۱، ۱۳۹ دستورالعمل یک بایتی، ۹۲ دستورالعمل دوبایتی و ۲۴ دستورالعمل سه بایتی دارد.
- CPU با استفده از چندین روش می تواند به داده ها دسترسی یابد:
  - داده مستقیماً در دستورالعمل قرار داده شده است
  - استفاده از روش های آدرس دهی مختلف برای داده ذخیره شده در حافظه کد و داده

روش های آدرس دهی

- هشت روش آدرس دهی عبارت است از:
  - آدرس دهی ثبات
  - آدرس دهی مستقیم
  - آدرس دهی غیر مستقیم
  - آدرس دهی فوری
  - آدرس دهی نسبی
  - آدرس دهی مطلق
  - آدرس دهی طولانی
  - آدرس دهی اندیس دار

آدرس دهی ثبات

- در استفاده از ثبات ها برای نگهداری داده کاربرد دارد.
- عملوند را در یک ثبات قرار می دهد و آن را با استفاده از استناد به ثبات (توسط اسم) در دستورالعمل مدیریت می کند.
  - mov A, R0 — مقدار R0 را به A منتقل می کند.
  - mov R2, A — مقدار A را به R2 منتقل می کند.
  - ADD A, R1 — مقدار A و مقدار R1 را جمع کرده در A قرار می دهد.
- ثبات های مبداء و مقصد می بایست از لحاظ اندازه همسان باشند.
- ممکن است دستورالعملی برای انتقال مقدار هر ثباتی به هر ثبات دیگر نباشد.
  - mov R4, R7 — نامعتبر است.
  - قبل از استفاده از دستورات لیست دستورات را چک کنید.

آدرس دهی مستقیم

- برای داده های ذخیره شده در RAM و ثبات ها استفاده می گردد.
  - تمامی مکان های حافظه با استفاده از آدرس ها قابل دسترسی هستند.
- از آدرس عملوند به صورت مستقیم در دستور استفاده می شود.
  - mov A, 40H حافظه [40H] را در A قرار می دهد. (بدون علامت # در قبل از 40H)
- آدرس دهی ثبات به صورت آدرس دهی مستقیم
  - mov A, 4H 4H آدرس ثبات R4 است.
  - mov A, R4 هر دو دستور مشابه هم هستند اما ممکن است کدهای عملیاتی متفاوتی داشته باشند
- تمامی ثبات ها و SFRها دارای آدرس هستند.
- پشته در ۸۰۵۱ تمها از روش آدرس دهی مستقیم استفاده می نماید.

آدرس دهی غیر مستقیم

- یک ثبات به عنوان یک اشاره گر مورد استفاده قرار می گیرد.
  - ثبات آدرس داده را در خود ذخیره می کند.
- تنها از R0، R1 و DPTR می توان در ۸۰۵۱ برای این منظور استفاده کرد.
- از R0 و R1 می توان برای حافظه داخلی (۲۵۶ بایت شامل SFRها) یا 00H تا FFH حافظه خارجی استفاده کرد.
  - mov A, @R0 حافظه داخلی [R0] را در A قرار می دهد.
  - mov @R1, A مقدار A را در حافظه داخلی [R1] قرار می دهد.
  - movx A, @R0 حافظه خارجی [R0] را در A قرار می دهد.
- DPTR می تواند برای اشاره به حافظه خارجی (تمامی 64K) مورد استفاده قرار بگیرد.
  - movx A, @DPTR حافظه خارجی [DPTR] را در A قرار می دهد.
  - movx @DPTR, A برعکس دستور فوق

آدرس دهی فوری

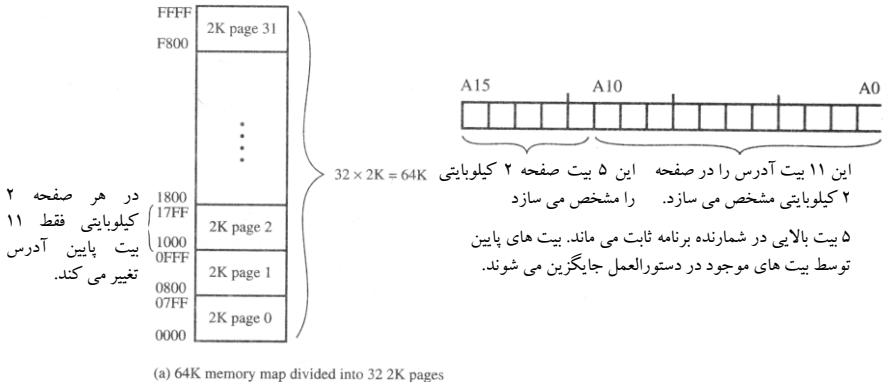
- عملوند (داده) به طور مستقیم در دستورالعمل (کد عملیاتی) تعیین می گرند.
- عملوند ثابتی است که در مدت زمان اسمبل آشکار می گردد.
- یک نشانه # در جلوی داده های فوری می آید.
  - mov A, #25 عدد ۲۵ را به انباره منتقل می کند.
  - mov DPTR, #1FFFH یک دستورالعمل سه بایتی که ثابت 1FFFH را به اشاره گر منتقل میکند

آدرس دهی نسبی

- تنها به وسیله چند دستورالعمل پرش مشخص مورد استفاده قرار می گیرد.
- آدرس نسبی یک عدد علامت دار ۸ بیتی است که با شمارنده برنامه جمع می شود و دستورالعمل بعدی که باید اجرا گردد را مشخص می سازد.
- پرش از روی ۱۲۸- تا ۱۲۷+ مکان امکان پذیر خواهد بود.
- پیش از جمع شدن شمارنده و فاصله نسبی، شمارنده برنامه یک واحد افزایش می یابد.
- چنانچه دستورالعمل SJMP 0107H در مکان های 0100H و 0101H قرار داشته باشد آنگاه فاصله نسبی از آدرس 0102H ، «۵» است.
- مقصد پرش معمولاً با یک برچسب مشخص می شود و اسمبلر مطابق با آن فاصله نسبی را تعیین می کند.

آدرس دهی مطلق

- تنها به وسیله دستورالعمل های ACALL و AJMP مورد استفاده قرار می گیرد.
- آدرس دارای ۱۱ بیت است. به هر خانه ای در صفحه ۲ کیلوبایت حاضر می تواند پرش کند.



(a) 64K memory map divided into 32 2K pages

آدرس دهی طولانی

- تنها به وسیله دستورالعمل های LJMPو LCALLمورد استفاده قرار می گیرد.
- دستورالعمل شامل یک آدرس ۱۶ بیتی کامل است.
- به هر خانه ای در حافظهK ۶۴ می تواند پرش کند.

آدرس دهی اندیس دار

- از یک ثبات برای ذخیره سازی اشاره گر و از ثبات دیگر برای افست استفاده می کند.
- آدرس موثر برابر مجموع پایه و افست می باشد.
- movc A, @A+DPTR حافظه کد خارجی [(A + DPTR)] به انباره منتقی می گردد.
- به طور وسیعی برای ایجاد جداول جستجو، آرایه های داده و ... در حافظه کد مورد استفاده قرار می گیرد.

انواع دستورالعمل ها

- دستورالعمل های ۸۰۵۱ به پنج گروه کاری تقسیم می شوند:
  - دستورالعمل های حسابی
  - دستورالعمل های منطقی
  - دستورالعمل های انتقال داده
  - دستورالعمل های متغیر بولی
  - دستورالعمل های انشعاب برنامه

دستورالعمل های حسابی

- چهار روش آدرس دهی امکان پذیر می باشد.
  - ADD A,7FH آدرس دهی مستقیم
  - ADD A,@R0 آدرس دهی غیر مستقیم
  - ADD A,R7 آدرس دهی ثبات
  - ADD A,#35H آدرس دهی فوری
- ADD: جمع ADDC: جمع با رقم نقلی SUB: تفریق SUBB:تفریق با رقم قرضی
  INC: افزایش DEC: کاهش MUL: ضرب DIV: تقسیم
- تمامی دستورالعمل های حسابی در یک سیکل ماشین اجرا می شوند. به غیر از INC DPTR (دو سیکل) و DIV AB ،MUL AB (چهار سیکل)

دستورالعمل های منطقی

- دستورالعمل های منطقی عملیات بولی را به صورت بیت به بیت بر روی بایت ها انجام می دهد.
- چهار روش آدرس دهی امکان پذیر می باشد.
  - ANL A,55H آدرس دهی مستقیم
  - ANL A,@R0 آدرس دهی غیر مستقیم
  - ANL A,R6 آدرس دهی ثبات
  - ANL A,#33H آدرس دهی فوری
- ANL: and منطقی ORL: or منطقی XRL: xor منطقی CLR پاک کردن
- CPL: متمم RL: چرخش به چپ RR: چرخش به راست SWAP: تعویض دو چهار بیت
- همه دستورالعمل های بولی که از انباره استفاده می کنند در یک سیکل و بقیه در دو سیکل اجرا می شوند.

دستورالعمل های انتقال داده

- **RAM داخلی**
  - دستورالعمل هایی که داده را در RAM داخلی جابجا می کنند در یک یا دوسیکل ماشین اجرا می شوند.
  - ۱۲۸ بایت بالای RAM داده تنها توسط آدرس دهی غیر مستقیم و SFRها با آدرس دهی مستقیم قابل دسترسی هستند.
  - PUSH و POP از آدرس دهی مستقیم استفاده می کنند اما خود پشته توسط آدرس دهی غیر مستقیم با استفاده از ثبات SP در دسترس خواهد بود.

دستورالعمل های انتقال داده

- **RAM خارجی**
  - دستورالعمل های انتقال داده بین حافظه داخلی و خارجی از آدرس دهی غیر مستقیم استفاده می کنند.
  - آدرس دهی غیر مستقیم یک بایتی با استفاده از @Ri که i=0,1 انجام شده و آدرس دهی غیر مستقیم دوبایتی با @DPTR مشخص می شود.
  - همه دستورالعمل های انتقال داده که با حافظه خارجی کار می کنند در دو سیکل ماشین اجرا می شوند.

دستورالعمل های انتقال داده

- **جدول های جستجو**
  - دو دستورالعمل انتقال داده برای خواندن جداول جستجوی واقع در حافظه وجود دارد.
  - نماد مورد استفاده برای انتقال ثابت MOVC می باشد.
  - از شمارنده برنامه یا اشاره گر داده به عنوان ثبات پایه و از انباره به عنوان فاصله نسبی استفاده می شود.
  - MOVC A,@A+DPTR

می تواند یک جدول با ۲۵۶ ورودی را از شماره ۰ تا ۲۵۶ را پشتیبانی نماید.

دستورالعمل های بولی

- حافظه RAM داخلی دارای ۱۲۸ بیت آدرس پذیر و فضای SFR هم تا ۱۲۸ بیت قابل آدرس دهی دیگر را پشتیبانی می کند.
- همه روش ها آدرس دهی به بیت از آدرس دهی مستقیم استفاده می کنند.
- CLR: پاک کردن SETB: یک کردن CPL: متمم کردن ANL: and کردن بیت
- ORL: or کردن بیت MOV: انتقال بیت JB: پرش در صورت یک بودن JNB

دستورالعمل های انشعاب برنامه

- دستورالعمل های زیادی برای کنترل روند برنامه از جمله فراخوانی و بازگشت زیر روال، انشعاب شرطی و غیر شرطی وجود دارد.
- سه روش آدرس دهی برای دستورالعمل های انشعاب برنامه وجود دارد.
- سه دستور JMP وجود دارد: SJMP (آدرس دهی نسبی) LJMP (آدرس دهی طولانی) AJMP (آدرس دهی مطلق)
- ACALL,LCALL: فراخوانی زیر روال RET: بازگشت از زیر روال RETI: بازگشت از وقفه
- JMP: پرش JZ: پرش اگر A صفر است. JNZ:پرش اگر Aصفر نیست
- CJNE: مقایسه و پرش در صورت مساوی نبودن DJNZ: کاهش و پرش اگر صفر نباشد
- NOP: انجام ندادن عملیات.