UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama
Sidang Akademik 1996/97

Oktober/November 1996

EEE 370 - Mikropemproses II

Masa : [2 jam]

ARAHAN KEPADA CALON :

Sila pastikan bahawa kertas peperiksaan ini mengandungi **LAPAN (8)** muka surat berserta LAMPIRAN (9) muka surat bercetak dan **ENAM (6)** soalan sebelum anda memulakan peperiksaan ini.

Jawab **EMPAT (4)** soalan.

Agihan markah bagi soalan diberikan di sut sebelah kanan soalan berkenaan.

Jawab semua soalan di dalam Bahasa Malaysia.

1. (i) Mikropengawal 'embedded' bagi aplikasi bolehgerak yang menggunakan bateri memerlukan mekanisma pengurusan kuasa yang efisien. Bincangkan ciri-ciri pengurusan kuasa yang terdapat dalam mikropengawal yang biasa digunakan. Huraikan contoh pengunaan salah satu ciri tersebut.

Embedded microcontrollers in mobile battery-powered applications require an efficient power management mechanism. Discuss the power management features available in typical microcontrollers, with an example application.

(20%)

(ii) Penjimatan kuasa adalah berkadaran dengan kekerapan mikropengawal dimasukkan ke dalam mod 'tidur' selama masa yang ditetapkan, apabila keperluan pemprosesan adalah rendah ataupun dihentikan buat sementara waktu. Tunjukkan bagaimana ini dilakukan bagi mikropengawal 8051

Power conservation is directly proportional to the number of times you can put a microcontroller to 'sleep' for a pre-determined period, when processing demand is low or temporarily terminated. Show how this is done for an 8051 microcontroller when

(a) mengunakan pemasa dalaman.
using the internal timers, and

(b) dengan litar 'wake-up' luaran.
with external 'wake-up' circuit.

Kod aturcara perlu diberikan.
Program codes must be given.

(40%)

(iii) Bagi kebanyakan aplikasi 'embedded', RAM mikropengawal mengandungi maklumat kritikal yang perlu disimpan sekiranya berlaku kehilangan bekalan kuasa. Huraikan dengan mendalam (dari segi perkakasan dan perisian) bagaimana ciri ini dilaksanakan jika peranti 87C51 digunakan dalam rekabentuk berkenaan.

For most embedded applications, the microcontroller RAM containing critical information must be saved in the event of power loss. Describe in detail (both hardware and software) how this feature can be implemented if an 87C51 microcontroller is used in the design.

(40%)

...3/-

2. Suatu sistem pengumpulan data 64-saluran 8-bit diperlukan bagi mengawasi parameter-parameter enjin bagi sebuah kapal besar. Adalah dikehendaki supaya beban pengumpulan data tersebut diagih-agihkan antara beberapa pemproses hamba, dan dikumpulkan oleh satu pemproses tuan, kesemuanya dihubungi oleh satu rangkaian multi-drop' RS-485. Pemproses tuan bolehlah seterusnya menghantarkan kesemua maklumat yang dikumpul ke satu komputer pusat untuk tujuan paparan, rekod dan analisis.

A 64-channel 8-bit data acquisition system is required for monitoring the engine parameters of a large ship. It is desired that the data acquisition tasks be distributed among several slave microcontrollers, and collected by one master unit, all linked by a multi-drop RS-485 network. The master can then pass on the collected data to a central PC for display, logging and analysis purposes.

(i) Berikan gambarajah blok bagi keseluruhan sistem tersebut, dengan huraian lengkap.

Produce a hardware block diagram for the overall system, with detailed description.

(30%)

(ii) Huraikan dengan mendalam perkakasan bagi modul mikropengawal hamba dan tuan.

Provide a complete hardware description of the slave and master microcontroller modules.

(30%)

(iii) Terangkan dengan lengkap bagaimana anda melaksanakan protokol perhubungan diantara modul-modul hamba dan tuan. Berikan contoh aturcara untuk memindahkan data dari modul hamba ke modul tuan.

Explain in detail how you would implement the communication protocol between the slaves and the master modules. Give an example program to transfer data from a slave to the master.

(40%)

...4/-

3. (a) Bincangkan masalah-masalah yang perlu dihadapi apabila merekabentuk sistem mikropengawal bagi kegunaan dalam persekitaran yang mempunyai bising elektrik.

Discuss the problems involved in designing microcontroller systems for applications in an electrically noisy environment.

Perkara-perkara berikut perlu dibincang dengan mendalam:
The following points must be discussed in detail:

(i) jenis dan sumber bising elektrik
types and sources of electrical noise

(ii) kesan bising
effects of noise

(iii) penangkisan
shielding

(iv) pembumian
grounds

(v) pengagihan bekalan kuasa dan 'decoupling'
power supply distribution and decoupling

(vi) strategi pembentangan PCB.
pcb layout strategy

(60%)

(b) Pemasa 'watchdog' adalah berguna untuk membolehkan pemproses pulih-semula dari kesilapan pelaksanaan aturcara. Walau bagaimanapun, tiada sekim 'watchdog' yang dapat memberi aras keyakinan 100%. Oleh itu, dalam aplikasi kritikal, bagaimanakah kebolehpercayaan 'watchdog' dapat ditingkatkan? (Berikan sekurang-kurangnya 4 teknik).

Watchdog timers are useful to help processors recover from software upsets. However, no watchdog scheme can provide 100% confidence. Therefore, in critical applications how can you increase the reliability of the watchdog itself? (Give at least 4 possible techniques).

(40%)

...5/-

4. Rekabentukkan suatu peranti pemasa berasaskan mikropengawal, yang mempunyai spesifikasi seperti berikut:

Design a microcontroller-based timer device, with the following specifications:

- 2 pemasa 'countdown', hanya satu aktif pada sesuatu ketika.
  2 independent countdown timer, but with only one being active at any particular time.
- Setiap pemasa membilang sebanyak 100 jam maksimum.
  Each timer can have a maximum count of 100 hours.
- Butang 'mula-henti' yang berasingan bagi setiap pemasa.
  A separate start-stop button for each timer.
- Baki masa boleh didapati bagi mana-mana pemasa, walaupun salah satu sedang aktif.
  A balance check can be done on any of the timers, even when one of them is counting.
- 'Alarm' dibunyikan apabila bilangan sifar dicapai oleh mana-mana pemasa.
  An alarm is sounded when zero count is reached by any of the timers.
- 'Keypad' diperlukan untuk mengsetkan pemasa.
  A keypad is required for setting the timer.
- LED 7-segmen digunakan sebagai paparan (JAM : MIN).
  7-segment LEDs are used for the display. (Hours : Min)
- Bekalan kuasa bateri
  Battery powered
- Data pemasa disimpan apabila peranti dimatikan.
  Timer data saved when device is off.
- Mempunyai antaramuka RS232
  RS232 serial interface

Maklumat berikut diperlukan:
The following information is required:

(i) Gambarajah blok bagi pemasa berkenaan, dengan penerangan lengkap bagaimana setiap fungsi dilaksanakan. Peranti-peranti yang digunakan juga perlu dijelaskan.

Block diagram of the proposed timer, with detailed explanation of how the functions are implemented. The devices to be used must be clearly identified.

(40%)

...6/-

(ii) Gambarajah skematik perkakasan.
Hardware schematic diagram (30%)

(iii) Carta alir bagi keseluruhan perisian.
Flowchart for the complete software. (30%)

5. Suatu peranti masukan perlu diantaramukakan ke pencetak siri. Peranti ini mempunyai satu talian masukan, RUN, yang apabila 'tinggi', mengaktifkan peranti berkenaan. Peranti tersebut kemudiannya mengeluarkan bait-bait data secara tak-segerak pada 8 talian DATA. Ianya juga mempunyai satu talian status data sah, DAV, yang menunjukkan data sah, setiap kali status berubah dari '1' ke '0'. Talian-talian RUN, DAV dan keluaran DATA sajalah merupakan sambungan kepada peranti tersebut. Masukan dan keluaran adalah TTL.

An input device is to be interfaced to a serial printer. The device has one input, RUN, which when high, operates the device. The device then outputs bytes of data asynchronously on its 8 data lines. It has one data valid status, DAV, which indicates valid data each time it makes a '1' to '0' transition. The RUN, DAV and data output are the only connection to the device. The input and output are TTL.

Antaramuka pencetak siri pula terdiri dari 3 talian, Rx, Tx dan GND. Ianya dikendalikan pada kadar 300 baud, tanpa jabat-tangan.

The printer's serial interface consist of 3 lines, i.e Rx, Tx and GND. It is to be operated at 300 baud, with no handshaking.

Terangkan bagaimana satu cip mikropengawal 8751 dapat dijadikan sebagai antaramuka diantara peranti masukan berkenaan dengan pencetak.

Show in detail how a single-chip 8751 microcontroller can be used as the interface between the device and the printer.

Perkara-perkara berikut perlu diberikan:

The following information is required:

...7/-

(i) Prinsip operasi, beserta gambarajah blok.
Detailed principle of operation, with block diagrams. (40%)

(ii) Skematik bagi antaramuka.
Complete schematic drawing for the interface. (30%)

(iii) Perisian pengawalan yang lengkap.
The complete control software, with comments. (30%)

6. Suatu peranti ingatan (X24C44) diantaramukakan ke mikropengawal 8051 seperti ditunjukkan dalam Rajah 1. Data untuk peranti berkenaan diberikan dalam Lampiran 2.

A memory device (X24C44) is interfaced to an 8051 microcontroller as shown in Figure 1. The data sheet for the device is given in Appendix 2.

(i) Dari data yang diberi, bincangkan kelebihan peranti ingatan jenis ini dan nyatakan beberapa kegunaan yang sesuai.

Briefly discuss the advantages of using this memory device and its possible applications.

(30%)

(ii) Terangkan bagaimana data ditulis ke lokasi RAM dalam peranti X24C44.

Explain how data is written to the RAM locations of the X24C44.

(30%)

(iii) Tuliskan aturcara 8051 untuk melaksanakan operasi tersebut.
Write an example 8051 program to perform this operation. (40%)

..8/-

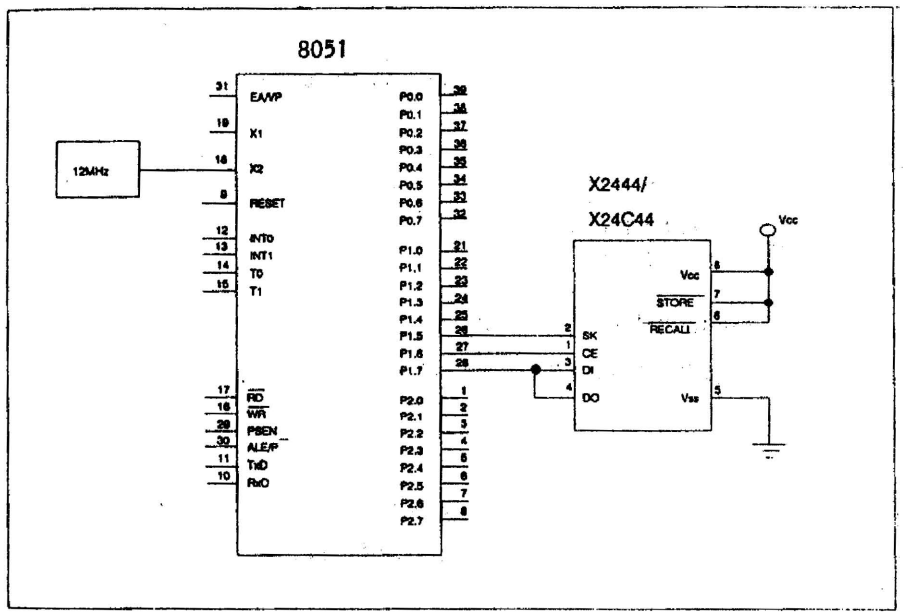


Figure 1

- oooOooo -

Table 8-10. 8051 Instruction Set Summary

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request, push the PC and to vector to the first instruction of the interrupt service program requires 38 to 86 oscillator periods (3 to 7 μs @ 12 MHz).

INSTRUCTIONS THAT AFFECT FLAG SETTINGS[1]

INSTRUCTION	FLAG			INSTRUCTION	FLAG		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:

- Rn —Register R7-R0 of the currently selected Register Bank.
- direct —8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
- @Ri —8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
- #data —8-bit constant included in instruction.
- #data 16 —16-bit constant included in instruction
- addr 16 —16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.
- addr 11 —11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
- rel —Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
- bit —Direct Addressed bit in Internal Data RAM or Special Function Register.
- • —New operation not provided by 8048AH/8049AH.

ARITHMETIC OPERATIONS

Mnemonic		Description	Byte	Oscillator Period
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,Rn	Subtract register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12

ARITHMETIC OPERATIONS Cont.

Mnemonic		Description	Byte	Oscillator Period
SUBB	A,@Ri	Subtract indirect RAM from Acc with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment indirect RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12

Table 8-10. 8051 Instruction Set Summary (Continued)

LOGICAL OPERATIONS

Mnemonic		Description	Byte	Oscillator Period
ANL	A,Rn	AND register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12

LOGICAL OPERATIONS Cont.

Mnemonic		Description	Byte	Oscillator Period
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12

Table 8-10. 8051 Instruction Set Summary (Continued)

DATA TRANSFER

Mnemonic		Description	Byte	Oscillator Period
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	Rn,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12

DATA TRANSFER Cont.

Mnemonic		Description	Byte	Oscillator Period
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12

Table 8-10. 8051 Instruction Set Summary (Continued)

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Oscillator Period
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to Carry	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24

PROGRAM BRANCHING

Mnemonic		Description	Byte	Oscillator Period
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24

PROGRAM BRANCHING Cont.

Mnemonic		Description	Byte	Oscillator Period
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	Rn,#data,rel	Compare immediate to register and Jump If Not Equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

APPLICATION NOTES
AVAILABLE
AN3 • AN7 • AN8 • AN15 • AN16 • AN25 • AN29 • AN30 • AN35 • AN36 • AN39 • AN56 • AN69

Xicor®

256 Bit X24C44 16 x 16 Bit

Serial Nonvolatile Static RAM

FEATURES

- Advanced CMOS Version of Xicor's X2444
- 16 x 16 Organization
- Single 5 Volt Supply
- Ideal for use with Single Chip Microcomputers
  - —Static Timing
  - —Minimum I/O Interface
  - —Serial Port Compatible (COPS™, 8051)
  - —Easily Interfaced to Microcontroller Ports
- Software and Hardware Control of Nonvolatile Functions
- Auto Recall on Power-Up
- TTL and CMOS Compatible
- Low Power Dissipation
  - —Active Current: 10mA Maximum
  - —Standby Current: 50µA Maximum
- 8-Pin Mini-DIP and 8-Lead SOIC Packages
- High Reliability
  - —Store Cycles: 1,000,000
  - —Data Retention: 100 Years

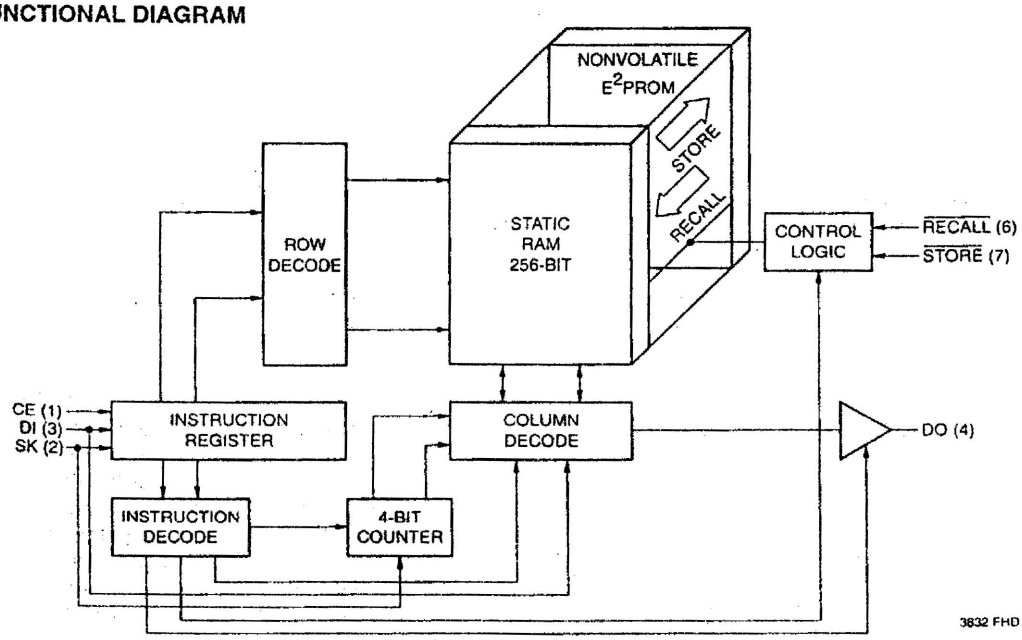
DESCRIPTION

The Xicor X24C44 is a serial 256 bit NOVRAM featuring a static RAM configured 16 x 16, overlaid bit-by-bit with a nonvolatile E²PROM array. The X24C44 is fabricated with Xicor's Advanced CMOS Floating Gate technology.

The Xicor NOVRAM design allows data to be transferred between the two memory arrays by means of software commands or external hardware inputs. A store operation (RAM data to E²PROM) is completed in 5ms or less and a recall operation (E²PROM data to RAM) is completed in 2µs or less.

Xicor NOVRAMs are designed for unlimited write operations to RAM, either from the host or recalls from E²PROM and a minimum 1,000,000 store operations. Inherent data retention is specified to be greater than 100 years.

FUNCTIONAL DIAGRAM



COPS is a trademark of National Semiconductor Corp.

X24C44

PIN DESCRIPTIONS

Chip Enable (CE)

The Chip Enable input must be HIGH to enable all read/write operations. CE must remain HIGH following a Read or Write command until the data transfer is complete. CE LOW places the X24C44 in the low power standby mode and resets the instruction register. Therefore, CE must be brought LOW after the completion of an operation in order to reset the instruction register in preparation for the next command.

Serial Clock (SK)

The Serial Clock input is used to clock all data into and out of the device.

Data In (DI)

Data In is the serial data input.

Data Out (DO)

Data Out is the serial data output. It is in the high impedance state except during data output cycles in response to a READ instruction.

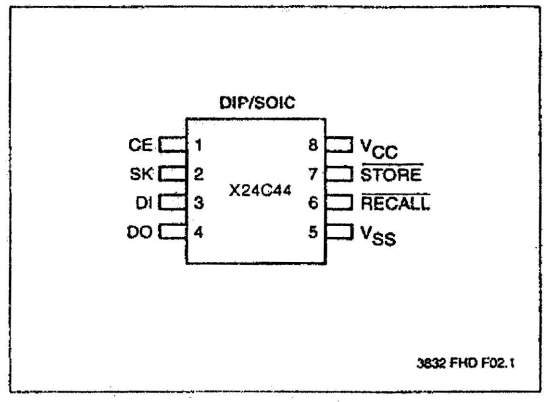
$\overline{\text{STORE}}$

$\overline{\text{STORE}}$ LOW will initiate an internal transfer of data from RAM to the E²PROM array.

$\overline{\text{RECALL}}$

$\overline{\text{RECALL}}$ LOW will initiate an internal transfer of data from E²PROM to the RAM array.

PIN CONFIGURATION



PIN NAMES

Symbol	Description
CE	Chip Enable
SK	Serial Clock
DI	Serial Data In
DO	Serial Data Out
$\overline{\text{RECALL}}$	Recall Input
$\overline{\text{STORE}}$	Store Input
V_{CC}	+5V
V_{SS}	Ground

3832 PGM T01

DEVICE OPERATION

The X24C44 contains an 8-bit instruction register. It is accessed via the DI input, with data being clocked in on the rising edge of SK. CE must be HIGH during the entire data transfer operation.

Table 1. contains a list of the instructions and their operation codes. The most significant bit (MSB) of all instructions is a logic one (HIGH), bits 6 through 3 are either RAM address bits (A) or don't cares (X) and bits 2 through 0 are the operation codes. The X24C44 requires the instruction to be shifted in with the MSB first.

After CE is HIGH, the X24C44 will not begin to interpret the data stream until a logic "1" has been shifted in on DI. Therefore, CE may be brought HIGH with SK running and DI LOW. DI must then go HIGH to indicate the start condition of an instruction before the X24C44 will begin any action.

In addition, the SK clock is totally static. The user can completely stop the clock and data shifting will be stopped. Restarting the clock will resume shifting of data.

RCL and $\overline{\text{RECALL}}$

Either a software RCL instruction or a LOW on the $\overline{\text{RECALL}}$ input will initiate a transfer of E²PROM data into RAM. This software or hardware recall operation sets an internal "previous recall" latch. This latch is reset upon power-up and must be intentionally set by the user to enable any write or store operations. Although a recall operation is performed upon power-up, the previous recall latch is not set by this operation.

WRDS and WREN

Internally the X24C44 contains a "write enable" latch. This latch must be set for either writes to the RAM or store operations to the E²PROM. The WREN instruction sets the latch and the WRDS instruction resets the latch, disabling both RAM writes and E²PROM stores, effectively protecting the nonvolatile data from corruption. The write enable latch is automatically reset on power-up.

STO and $\overline{\text{STORE}}$

Either the software STO instruction or a LOW on the $\overline{\text{STORE}}$ input will initiate a transfer of data from RAM to E²PROM. In order to safeguard against unwanted store operations, the following conditions must be true:

- STO instruction issued or $\overline{\text{STORE}}$ input is LOW.
- The internal "write enable" latch must be set (WREN instruction issued).
- The "previous recall" latch must be set (either a software or hardware recall operation).

Once the store cycle is initiated, all other device functions are inhibited. Upon completion of the store cycle, the write enable latch is reset. Refer to Figure 4 for a state diagram description of enabling/disabling conditions for store operations.

WRITE

The WRITE instruction contains the 4-bit address of the word to be written. The write instruction is immediately followed by the 16-bit word to be written. CE must remain HIGH during the entire operation. CE must go LOW before the next rising edge of SK. If CE is brought LOW prematurely (after the instruction but before 16 bits of data are transferred), the instruction register will be reset and the data that was shifted-in will be written to RAM.

If CE is kept HIGH for more than 24 SK clock cycles (8-bit instruction plus 16-bit data), the data already shifted-in will be overwritten.

Table 1. Instruction Set

Instruction	Format, I_2 I_1 I_0	Operation
WRDS (Figure 3)	1XXXX000	Reset Write Enable Latch (Disables Writes and Stores)
STO (Figure 3)	1XXXX001	Store RAM Data in E²PROM
Reserved	1XXXX010	N/A
WRITE (Figure 2)	1AAAA011	Write Data into RAM Address AAAA
WREN (Figure 3)	1XXXX100	Set Write Enable Latch (Enables Writes and Stores)
RCL (Figure 3)	1XXXX101	Recall E²PROM Data into RAM
READ (Figure 1)	1AAAA11X	Read Data from RAM Address AAAA

3832 PGM T13

X = Don't Care
A = Address

X24C44

READ

The READ instruction contains the 4-bit address of the word to be accessed. Unlike the other six instructions, I_0 of the instruction word is a "don't care". This provides two advantages. In a design that ties both DI and DO together, the absence of an eighth bit in the instruction allows the host time to convert an I/O line from an output to an input. Secondly, it allows for valid data output during the ninth SK clock cycle.

D0, the first bit output during a read operation, is truncated. That is, it is internally clocked by the falling edge of the eighth SK clock; whereas, all succeeding bits are clocked by the rising edge of SK (refer to Read Cycle Diagram).

LOW POWER MODE

When CE is LOW, non-critical internal devices are powered-down, placing the device in the standby power mode, thereby minimizing power consumption.

SLEEP

Because the X24C44 is a low power CMOS device, the SLEEP instruction implemented on the first generation NMOS device has been deleted. For systems converting from the X2444 to the X24C44 the software need not be changed; the instruction will be ignored.

WRITE PROTECTION

The X24C44 provides two software write protection mechanisms to prevent inadvertent stores of unknown data.

Power-Up Condition

Upon power-up the "write enable" latch is in the reset state, disabling any store operation.

Unknown Data Store

The "previous recall" latch must be set after power-up. It may be set only by performing a software or hardware recall operation, which assures that data in all RAM locations is valid.

SYSTEM CONSIDERATIONS

Power-Up Recall

The X24C44 performs a power-up recall that transfers the E²PROM contents to the RAM array. Although the data may be read from the RAM array, this recall does not set the "previous recall" latch. During this power-up recall operation, all commands are ignored. Therefore, the host should delay any operations with the X24C44 a minimum of t_{PUR} after V_{CC} is stable.

Power-Down Data Protection

Because the X24C44 is a 5V only nonvolatile memory device it may be susceptible to inadvertent stores to the E²PROM array during power-down cycles. Power-up cycles are not a problem because the "previous recall" latch and "write enable" latch are reset, preventing any possible corruption of E²PROM data.

Software Power-Down Protection

If the $\overline{\text{STORE}}$ and $\overline{\text{RECALL}}$ pins are tied to V_{CC} through a pull-up resistor and only software operations are performed to initiate stores, there is little likelihood of an inadvertent store. However, if these two lines are under microprocessor control, positive action should be employed to negate the possibility of these control lines bouncing and generating an unwanted store. The safest method is to issue the WRDS command after a write sequence and also following store operations. Note: an internal store may take up to 5ms; therefore, the host microprocessor should delay 5ms after initiating the store prior to issuing the WRDS command.

Hardware Power-Down Protection

(when the "write enable" latch and "previous recall" latch are not in the reset state):

Holding either $\overline{\text{RECALL}}$ LOW, CE LOW or $\overline{\text{STORE}}$ HIGH during power-down will prevent an inadvertent store.

Figure 1. RAM Read

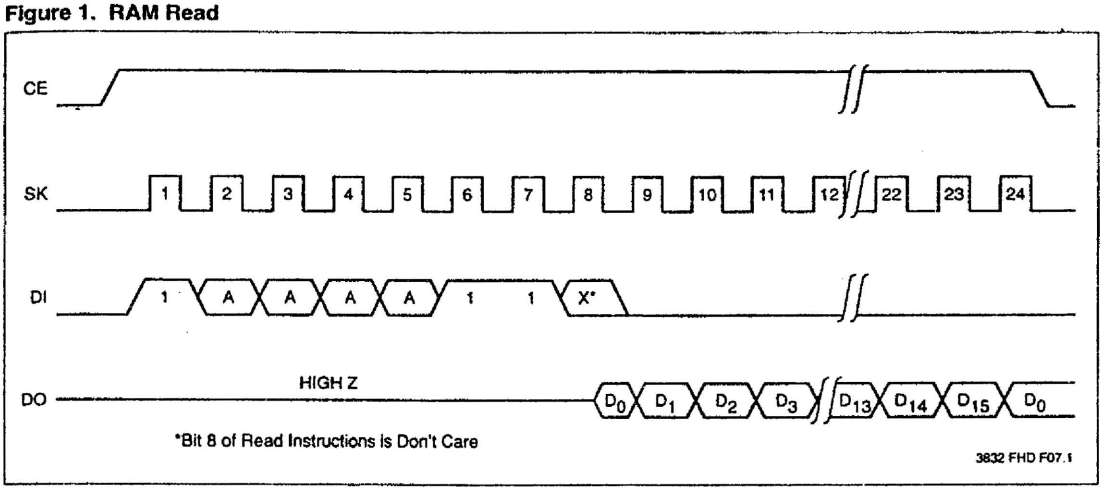


Figure 2. RAM Write

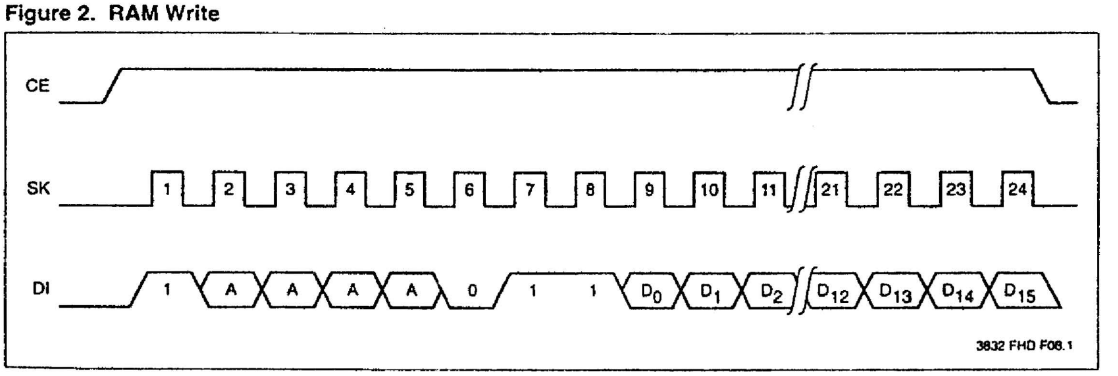


Figure 3. Non-Data Operations