LU800V1/LU800AV1/LU805BV2

CMOS 8Bit Single Chip Microcomputers (ROM less)

■ Description

The LU800V1/LU800AV1/LU805BV2 is a ROMless version of the SM803/A and SM805/A CMOS 8-bit single-chip microcomputers and offers the outstanding feature of the Z8 family architecture.

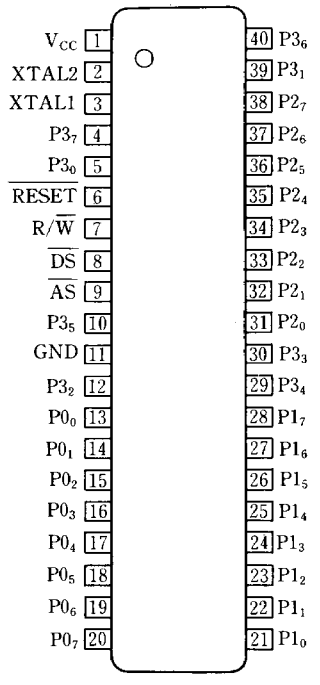
Because some I/O ports are used for address/data bus, this device accesses up to 128K bytes of the external memory space. Using the external memory in place of an on-chip ROM allows designing more powerful microcomputer system.

■ Features

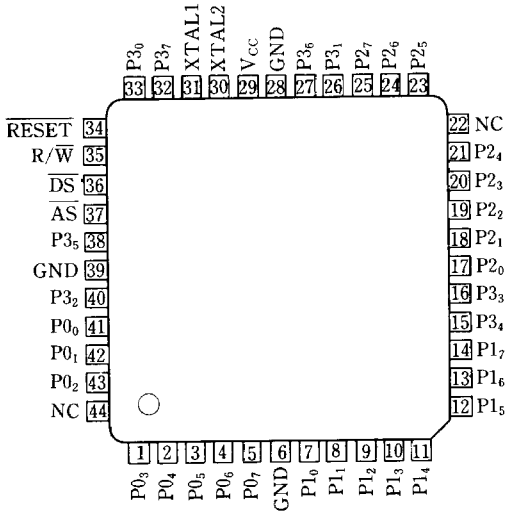
1. Complete microcomputer, 24 I/O lines, and up to 64K bytes addressable external space each for program and data memory.
2. 143 bytes register file
   (255 bytes register file for the LU805BV2)
   124 general-purpose registers
   (236 registers for the LU805BV2)
   3 I/O port registers
   16 status and control registers
3. Register pointer so that short, fast instructions can access any one of the 9 working-register groups.
   (16 groups for the LU805BV2)

■ Pin Connections

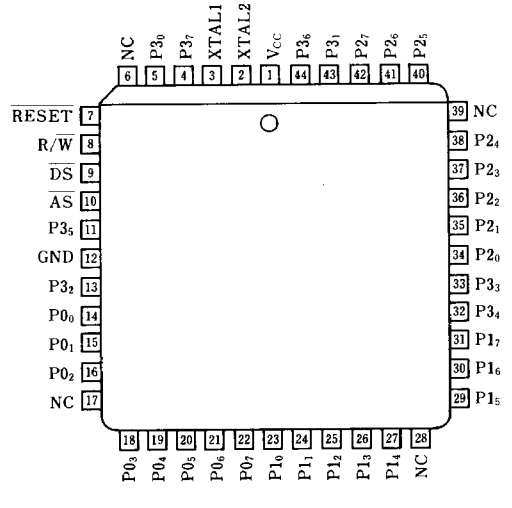
LU800V1/LU800AV1/LU805BV2



LU800V1M/LU800AVM/LU805BVM



LU800V1U/LU800AVU/LU805BVU

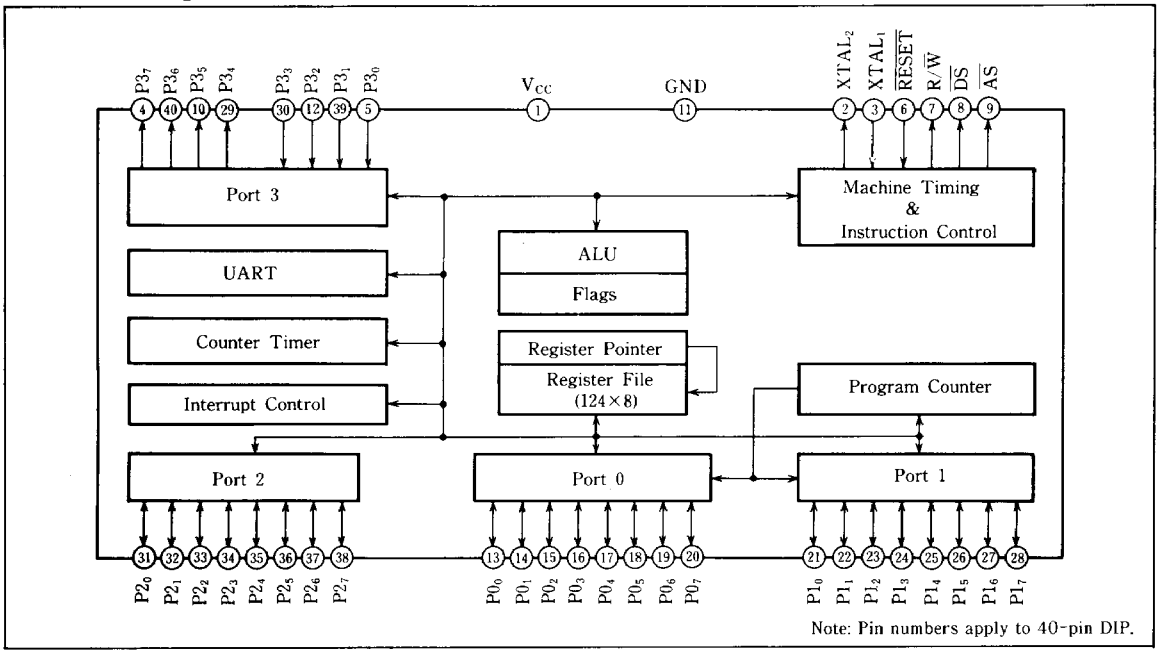


4. Full-duplex UART and two programmable 8-bit counter/timers, each with a 6-bit programmable prescaler.
5. Vectored priority interrupts for I/O, counter/timers, and UART.
6. On-chip oscillation circuit
7. External clock
   8MHz MAX. (internal 4MHz): LU800V1/M
   12MHz MAX. (internal 6MHz): LU800AV1/M
   16MHz MAX. (internal 8MHz): LU805BV2/M
8. Single +5V power supply
9. 40-pin DIP (DIP40-P-600)
   LU800V1/LU800AV1/LU805BV2
   44-pin QFP (QFP44-P-1414)
   LU800V1M/LU800AVM/LU805BVM
   44-pin QFJ (QFP44-P-S650)
   LU800V1U/LU800AVU/LU805BVM

■ Ordering Information

Model No.	Clock	Package
LU800V1	8MHz	40DIP
LU800V1M		44QFP
LU800V1U		44QFJ
LU800AV1	12MHz	40DIP
LU800AVM		44QFP
LU800AVU		44QFJ
LU805BV2	16MHz	40DIP
LU805BVM		44QFP
LU805BVU		44QFJ

■ Block Diagram



Note: Pin numbers apply to 40-pin DIP.

■ Pin Description

Pin	Meaning	I/O	Function
$P0_0$-$P0_7$	Port 0	I/O	8-bit I/O port, programmable for I/O.
$P1_1$-$P1_7$	Address/data bus	I/O	Multipiexed Address/data bus
$P2_0$-$P2_7$	Port 2	I/O	Programmable for I/O in bits.
$P3_0$-$P3_7$	Port 3	I/O	$P3_0$-$P3_3$ for input, $P3_4$-$P3_7$ for output.
$\overline{AS}$	Address Strobe	O	Active "Low", activated for external address memory transfer.
$\overline{DS}$	Data Strobe	O	Active "Low", activated for external data memory transfer.
R/$\overline{W}$	Read/Write	O	Read at "High", Write at "Low".
$\overline{RESET}$	Reset	I	Active "Low", Initializes.
XTAL1	Clock 1	I	Clock terminal pin.
XTAL2	Clock 2	O	Clock terminal pin.

Pin functions of the LU800V1/LU800AV1/LU805BV2 are identical to those of the SM803/A, SM805/A, except for pins $P1_0$-$P1_7$.

■ Address space

(1) Program Memory

The ROMless device, having a 16-bit program counter, addresses 64K-bytes of external program memory. All the command codes are fetched from these external program memories.

For the ROMless device, the first 12 bytes of program memory are reserved for the interrupt vectors. These locations contain six 16-bit vectors that correspond to the six available interrupts. Program execution begins at location $000C_H$ after a resed.

(2) Data Memory*

The ROMless device can address 64K bytes of external data memory. External data memory may be included with or separated from the external program memory space. $\overline{DM}$, an optional I/O function that can be programmed to appear on pin $P3_1$, is used to distinguish between data and program memory space:

(3) Register File

The 143-byte register file inculdes three I/O port registers (R0, R2, R3), 124 general-purpose registers (R4-R127) and 16 control and status registers (R240-R255).

These registers are assigned the address locations shown in Fig 2.

The instructions can access registers directly or indirectly with an 8-bit address field. This also allows short 4-bit register addressing using the Register Pointer (one of the control registers). In the 4-bit mode. the regiter file is divided into nine working register groups, each occupying 16 contiguous locations. The Register Pointer addresses the starting location of the active working-register group.

(4) Stacks

Either the internal register file or the external data memory can be used for the stack. A 16-bit Stack Pointer (R254 and R255) is used for the external stack, which can reside anywhere in data memory. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 124/236 for the LU805BV2 general-purpose registers (R4-R127/R4-R239).

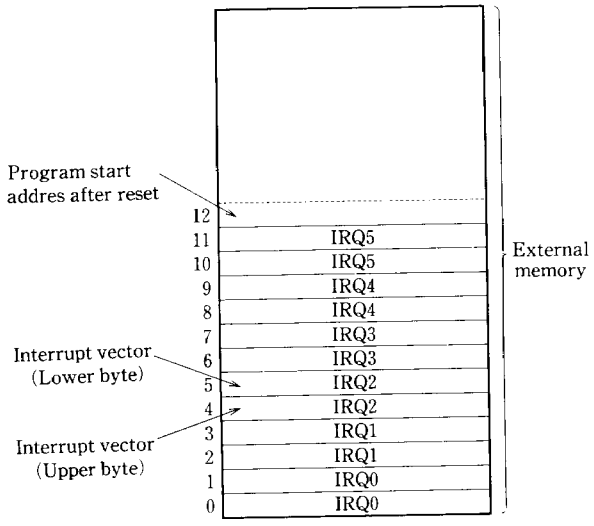


Fig. 1 Program memory map

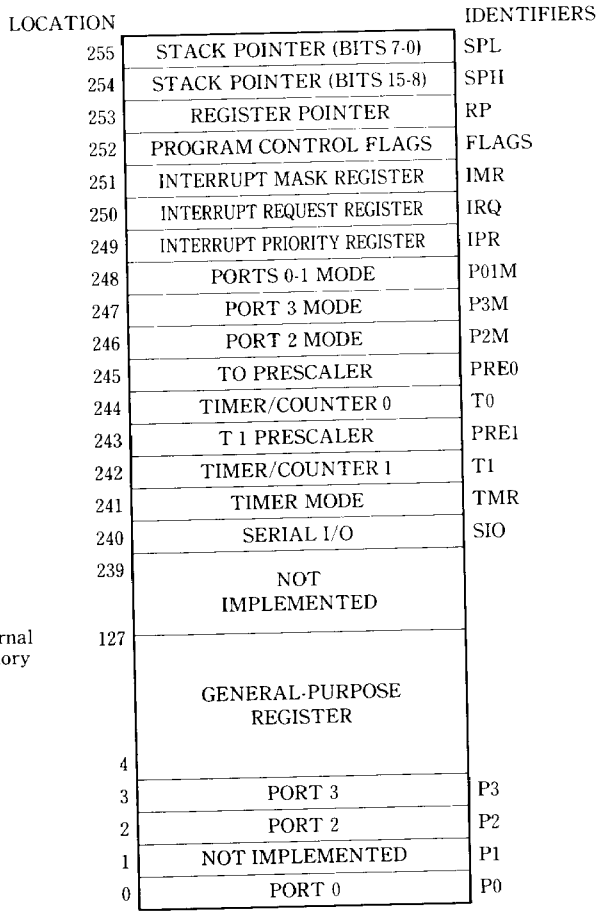


Fig. 2 The register file

Port Functions

The LU800V1/LU800AV1/LU805BV2 has a dedicated memory interface port (Port 1) and input/output ports (Port 0, 2, 3). These ports are given eight lines each. The functions of port 0, 2 and 3 are the same as those of the SM803/A, SM805/A.

Port 1 is a dedicated Z-bus compatible memory interface. The operations of Port 1 are supported by the Address Strobe ($\overline{AS}$) and Data Strobe ($\overline{DS}$) lines, and by the Read/Write ($R/\overline{W}$) and Data Memory ($\overline{DM}$) control lines.

The low-order program and data memory address (A_0-A_7) are output through Port 1 and are multiplexed with data in/out (D_0-D_7). Instruction fetch and data memory read/write operations are done through this port.

Port 1 cannot be used as a register nor can a handshake mode be used with this port.

If more than address lines are required with the ROMless device, additional lines can be obtained by programming Port 0 bits as address bits. The least significant four bits of Port 0 can be configured to supply address bits A_8-A_{11} for 4K byte addressing or both nibbles of Port 0 can be configured to supply address bits A_8-A_{15} for 64K byte addressing.

Registers

The LU800V1/LU800AV1/LU805BV2 control registers are the same as on the SM803/A, SM805/A, except two bits D_3 and D_4 in the port 0, 1 mode register (R248).

Serial Input/Output

The LU800V1/LU800AV1/LU805BV2 serial input/output functions are the same as those of the SM803/A, SM805/A, (Refer back to the SM803/A, SM805/A description.)

Counter/Timers

The LU800V1/LU800AV1/LU805BV2 counter/timer functions are the same as those of the SM803/A, SM805/A, (Refer back to the SM803/A, SM805/A description.)

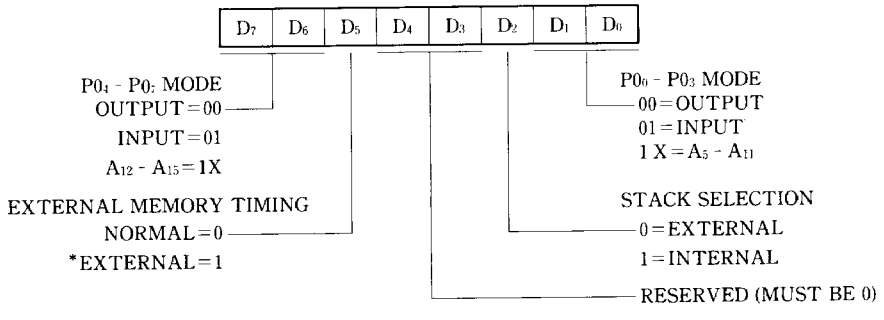
Interrupts

The LU800V1/LU800AV1/LU805BV2 interrupt functions are the same as those of the SM803/A, SM805/A, (Refer back to the SM803/A, SM805/A description.)

Instructions and AC/DC Characteristics

These data of the LU800V1/LU800AV1/LU805BV2 are the same as for the SM803/A, SM805/A. (Refer back to the SM803/A, SM805/A description.)

R248 (P01M) Port 0, 1 Mode Register ($F8_H$ Write only)



■ Reset

When the NU800V1 is reset, the device must be kept Low for at least 50msec from the device is stabled with the reset switch is turned on, or for 18 clock cycles from the power supply and clock oscillator are stabled.

The intervals reset the LU800V1 is obtained by connecting external capacitor of 1μF and resistor of 100kΩ as shown in Fig. 3.

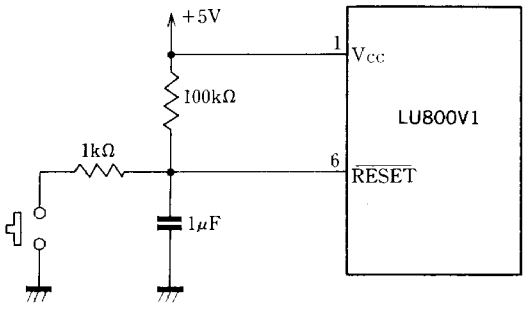


Fig. 3

After reset, ports 0 and 2 are used as input ports, the program counter is reset at 000_H and the interputs are disabled.

When the reset input inactivated, the program memory starts execution at $000C_H$.

■ Initialization

When the program, after reset, starts execution at $000C_H$, the device must be initialized. Ports 0 and 2 after reset are used as inputs, and an expanded memory timing and an internal stack are selected with the $\overline{DM}$ signal not to be output. The valid address lines include 8 lines of port 1 only. Usable memory sholud be limited to the first 256 bytes and the port 0 must be programmed as address lines within 256 bytes of memory for use of more than 257 bytes of memory.

Port 0, $P0_0$-$P0_7$, is used as input port after reset, if is used as address lines, a constant address value must be held with an external circuit until it is initialized.

The port initialization sequence is:

(1) Write upper byte of address of an initialization routine to port 0 register.

(2) Configure port 0 and 1 mode register P01M. ($D_1=1$; lower 4 bits of port 0 should be A_8-A_{11} address lines, $D_7=1$ every bit of port 0 should be A_8-A_{15} address lines.)

Note: While the next byte of instruction indicated in the above item (2) are being fetched, be sure not to make a difference between an address output from port 0 and an address held in an external circuit. (This is because the instruction is executed in pipeline system.)

Initiflization with Pull-up Resistors

When connecting the lower bits $P0_0$-$P0_3$ of port 0 to a 4K byte memory with a pull-up resistor, Fig.4 shows that the addresses A_8-A_{11} are kept during the port 0 is an unknown state, and must be phisically located in the latter address of $F0C_H$ of a 4K byte memory. The A_8-A_{11} will change according to the address output from port $P0_0$-$P0_3$, if the port 0 is used as address lines.

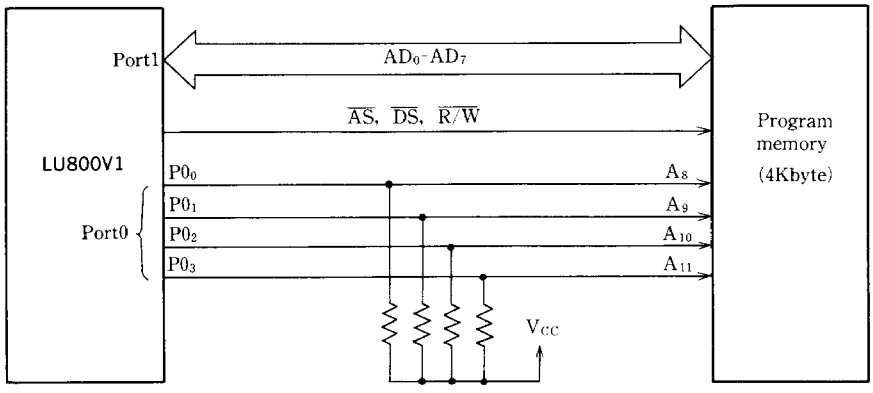


Fig. 4 Memory interface with pull-up resistors

Initialization of port 0

(1) Jump to the address FXX_H in order to match the program counter to the address being accessed.
(2) Write $0F_H$ (upper byte of the address)into port 0 register at FXX_H.
(3) Set the proper bits in the port 0 and 1 mode register to output the port 0.
(4) Set the proper bits in the port 0 and 1 mode register to use the port 0 as address lines.

Initialization with The LS157

Fig. 5 shows the memory interface between upper 4 bits ($P0_0$-$P0_3$) of port 0 and a 4K byte memory with the LS157 and a flip-flop.

After reset, in this case, the "b" inputs are selected and address bits A_8-A_{11} go Low because the SELECT input to the LS157 is kept High until the R/$\overline{W}$ goes Low. If the R/$\overline{W}$ goes Low, the SELECT goes Low and $P0_0$-$P0_3$ will be valid.

Initialization of port 0

(1) Write 00_H (upper byte address of initialization routine) to port 0 register.
(2) Set the proper bits in the port 0 and 1 mode registers.
(3) Write into the external memory upon execution of an LDC or LDE intruction. (This allows the R/$\overline{W}$ to go Low and the LS157 to switch to the "a" inputs.

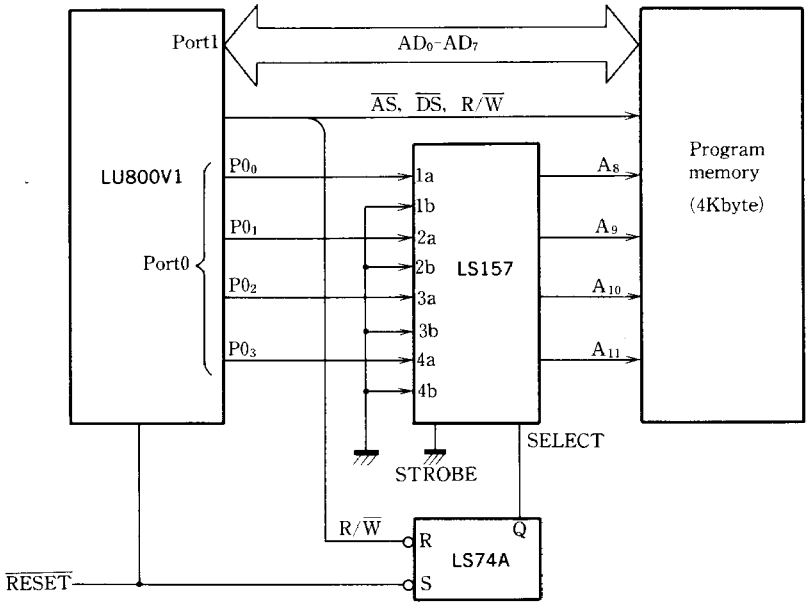


Fig. 5 Memory interface with the LS157