Microcomputer for multi-key remote controllers

BU2456/BU2458/BU2459/BU2460/BU2461/BU2462

The BU2458 series are four-bit, single-chip microcomputers on which all of the functions required for a remote control transmitter can be configured on a single chip, which can be used as a microcontroller for a remote control encoder.
These are ideal for keeping system sizes compact, keeping prices low, and building sophisticated functions.

●Applications
Remote control encoders

Specification

Series	BU2456/BU2458/BU2459/BU2460	BU2461/BU2462
Program memory (ROM) (bytes)	1024	1024
Data memory (RAM) (bits)	32×4	32×4
Subroutine nesting levels	3	3
Instruction sets	43	42
Large-current output port	1	1
Input ports	4	4
Output ports	8	8
I/O ports	4	2
Instruction cycle (μs)	13.2 (f_{OSC}=455kHz)	13.2 (f_{OSC}=455kHz)
Power supply voltage (V) typ.	3	3

●Features
1) Five types of internal carrier generator circuits for infrared remote control signals (selectable via program)
2) Internal capacitor for ceramic oscillation circuit (mask option)
3) Internal watchdog timer
4) Internal Power On Reset function
5) Large-current output port (for remote control signal output)
6) Clock frequency : 300kHz ~ 1MHz
7) Program memory (ROM) of 1024 bytes
8) Data memory (RAM) of 32 × 4 bits
9) 3 levels of subroutine nesting
10) 4-bit input ports (for key scanning)
11) 4-bit input/output ports (for key scanning and expansion pins)
12) 8-bit individual output ports (for key scanning)
13) Instruction cycle of 13.2 μs (when f_{OSC} = 455kHz)
14) HALT function can be cancelled through key input

●Absolute maximum ratings (Ta=25℃)

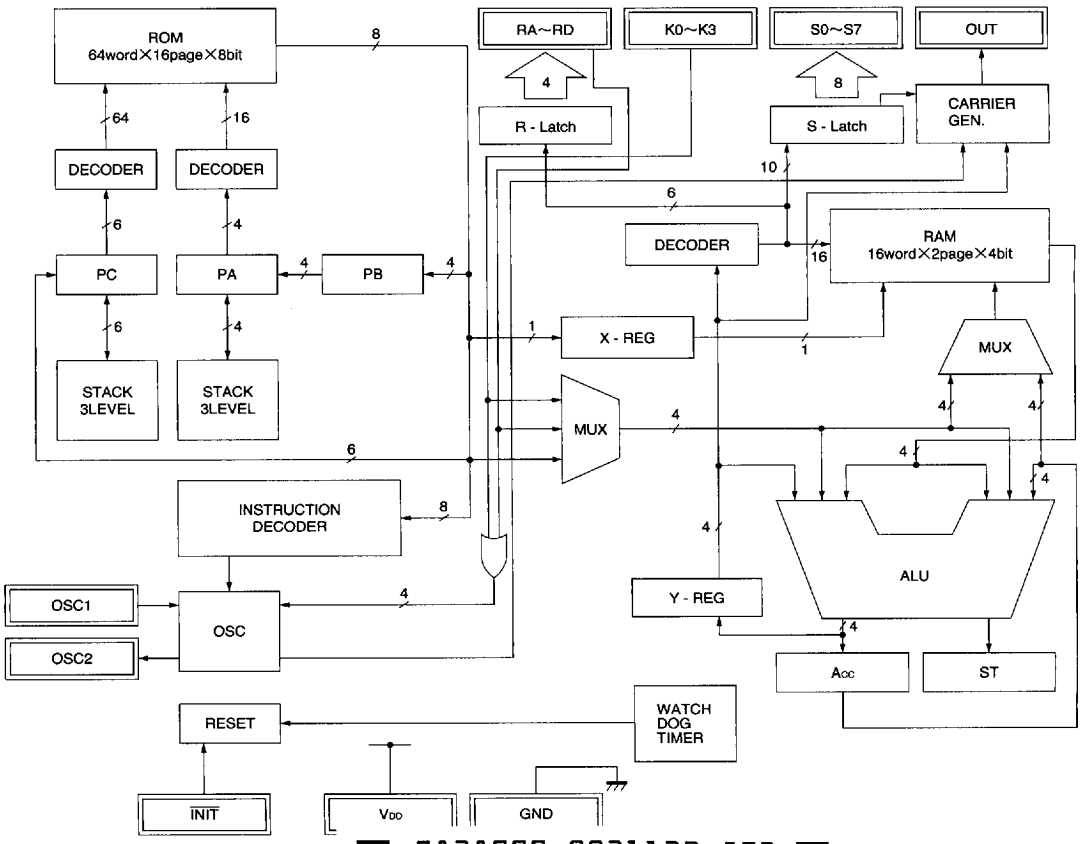
Parameter	Symbol	Limits	Unit
Power supply voltage	V_{DD}	−0.3～5.0	V
Power dissipation	P_d	550*1 (BU2458/BU2461) 600*2 (BU2459/BU2460/BU2462) 800*3 (BU2456)	mW
Storage temperature	T_{stg}	−55～+125	℃
Input voltage	V_{IN}	−0.3～V_{DD}+0.3	V
Output voltage	V_{OUT}	−0.3～V_{DD}+0.3	V

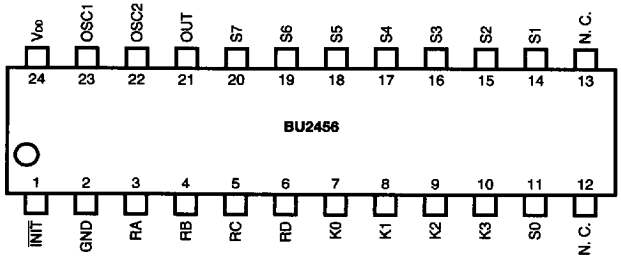
*1 Reduced by 5.5mW for each increase in Ta of 1℃ over 25℃.
*2 Reduced by 6mW for each increase in Ta of 1℃ over 25℃.
*3 Reduced by 8.0mW for each increase in Ta of 1℃ over 25℃.

●Recommended operating conditions

Parameter	Symbol	Limits	Unit
Power supply voltage	V_{DD}	2.0～4.0	V
Operating temperature	T_{opr}	−25～+75	℃

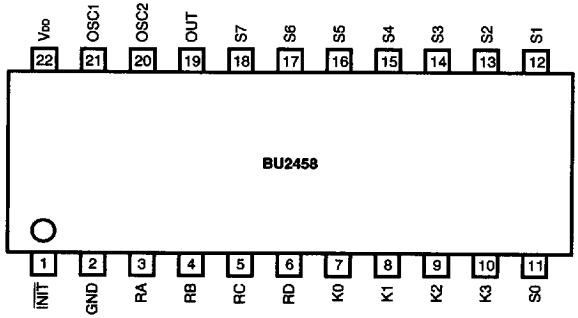
●Block diagram



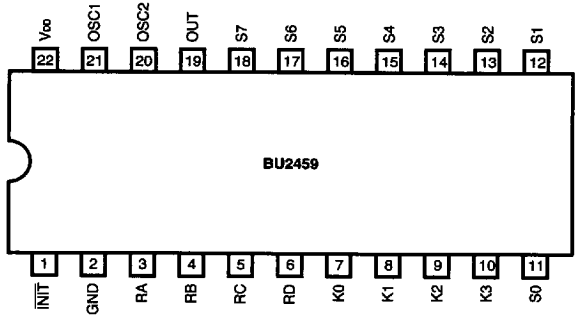


N.C.: Not connected

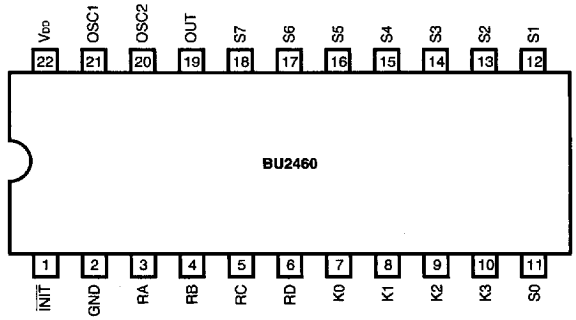
BU2456 pin layout



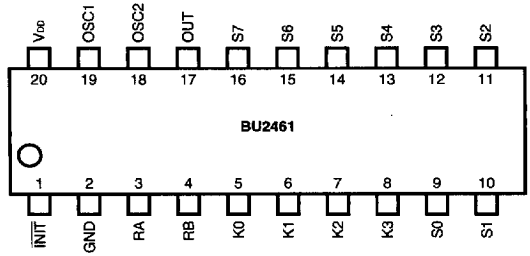
BU2458 pin layout



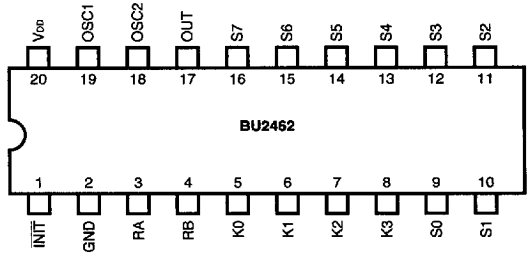
BU2459 pin layout



BU2460 pin layout



BU2461 pin layout



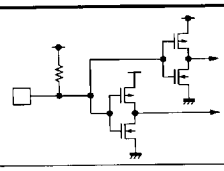
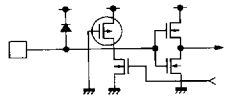
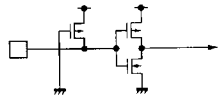
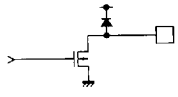
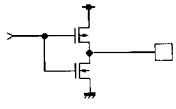
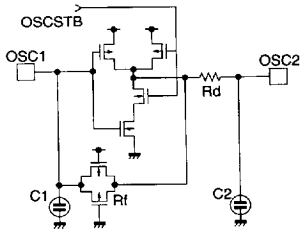
BU2462 pin layout

●Pin description

Pin Name	I/O	Function
V_{DD}	—	Used to connect 2.0 V ~ 4.0 V power supply.
GND	—	Reference voltage for all inputs and outputs (0 V).
$\overline{INIT}$	Input	Manual reset input. Setting this pin to LOW initializes the S output ports and OUT output ports to LOW and sets the ROM address to 0 page, 0 address.
K0~K3	Input	4-bit input ports. Internal pull-up resistance. A mask option can be used to select whether or not the HALT cancel function is to be effective for each individual bit (when the HALT function is effective, the HALT status is cancelled by setting this to LOW).
S0~S7	Output	Each of these can be set and reset independently (or all at once). The output format is Nch open drain.
RA~RD	Input/output	4-bit input/output ports (however, these are set in input mode if a HIGH state is output for any respective pin). The outputs can be set and reset independently (or all at once). The output format is Nch open drain. A mask option can be used to selected whether or not the HALT cancel function is to be effective (by setting this to LOW) and whether pull-up resistance is used.
OUT	Output	Remote control signal output port which can be used to drive large currents. The output format is CMOS output. The output HIGH current side can be used to drive large currents.
OSC1	Input	A ceramic resonator is connected between this pin and OSC2. Built-in feedback resistor between this pin and OSC2.
OSC2	Output	A ceramic resonator is connected between this pin and OSC1.

Note: There are no RC or RD pins in the BU2461 and BU2462.

●Input/output circuit

Pin Name	I/O	I/O Circuit	Notes
$\overline{\text{INIT}}$	Input/output		Hysteresis input Internal pull-up resistance About. 400 kΩ
RA~RD	Input/output		Open drain output HIGH on reset (optional) Internal MOS Tr for pull-up About. 120 kΩ
K0~K3	Input		Internal MOS Tr for pull-up About. 120 kΩ
S0~S7	Output		Open drain output LOW on reset
OUT	Output		CMOS output LOW on reset Output HIGH current side can drive large currents
OSC2	Output		Internal MOS Tr for feedback Rf: About. 1 MΩ Internal damping resistance Rd: About. 6 kΩ
OSC1	Input		(optional) Internal capacitors can be provided for oscillation C1: About. 100 pF C2: About. 100 pF

Note: There are no RC or RD pins in the BU2461 and BU2462.

○ : Circled items are mask options.

●Electrical characteristics (Unless otherwise noted, Ta=25℃, V_{DD}=3V)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Conditions
$\overline{INIT}$ input current (HIGH)	I_{IHIT}	—	—	1	μA	$V_I=V_{DD}$
$\overline{INIT}$ input current (LOW)	I_{ILIT}	−3	−7.5	−16	μA	V_I=GND
K input current (HIGH)	I_{IHK}	—	—	1	μA	$V_I=V_{DD}$
K input current (LOW)	I_{ILK}	−9	−25	−50	μA	V_I=GND
K input voltage (HIGH)	V_{IHK}	2.1	—	3	V	—
K input voltage (LOW)	V_{ILK}	0	—	0.9	V	—
R input current (HIGH)	I_{IHR}	—	—	1	μA	$V_I=V_{DD}$
R input current 1 (LOW)	I_{ILR1}	—	−0.1	−1	μA	V_I = GND, output OFF, no pull-up resistance
R input current 2 (LOW)	I_{ILR2}	−9	−25	−50	μA	V_I = GND, output OFF, no pull-up resistance
R input voltage (HIGH)	V_{IHR}	2.1	—	3	V	—
R input voltage (LOW)	V_{ILR}	0	—	0.9	V	—
$\overline{INIT}$ input voltage (HIGH)	V_{IHI}	2.25	—	3	V	—
$\overline{INIT}$ input voltage (LOW)	V_{ILI}	0	—	0.75	V	—
R input voltage (LOW)	V_{OLR}	—	0.15	0.4	V	I_{OL}=1mA
S input voltage (LOW)	V_{OLS}	—	0.15	0.4	V	I_{OL}=1mA
OUT output voltage (LOW)	V_{OLOT}	—	0.15	0.4	V	I_{OL}=100 μA
OUT output voltage (HIGH)	V_{OHOT}	2.1	2.5	—	V	I_{OH}=−8mA
OSC2 output voltage (LOW)	V_{OLOS}	—	0.4	0.9	V	I_{OL}=70 μA
OSC2 output voltage (HIGH)	V_{OHOS}	2.1	2.5	—	V	I_{OH}=−70 μA
R output leakage current	I_{LR}	—	—	1	μA	V_O = V_{DD}, output OFF
S output leakage current	I_{LS}	—	—	1	μA	V_O = V_{DD}, output OFF
OSC1 feedback current	I_{OSC1}	1	3	7	μA	When V_{OSC1} = GND, V_{OSC2} = V_{DD}, in HALT mode
Static current consumption	I_{DDST}	—	—	1	μA	In HALT mode
Operating current consumption 1	I_{DDOP1}	—	0.3	1.0	mA	When f_{OSC} = 455 kHz
Operating current consumption 2	I_{DDOP2}	—	0.25	—	mA	External clock of 1 MHz
Operating frequency	f_{OSC}	300	—	1000	kHz	—
Oscillation capacitance 1	C1	—	100	—	pF	When using internal capacitor
Oscillation capacitance 2	C2	—	100	—	pF	When using internal capacitor

●Measurement circuits

① Input current (HIGH) measurement circuit
② Input current (LOW) measurement circuit

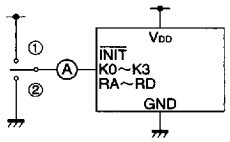


Fig. 1

① S output, R output voltage (LOW) measurement circuit
② OSC2 output voltage (LOW) measurement circuit
③ OUT output voltage (LOW) measurement circuit
(When testing, Nch transistor should be on.)

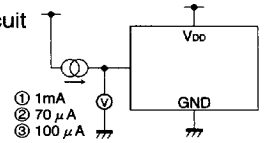


Fig. 2

① OUT output voltage (HIGH) measurement circuit
② OSC2 output voltage (HIGH) measurement circuit
(When testing, Pch transistor should be on.)

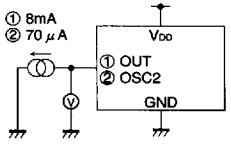


Fig. 3

R output, S output leakage current measurement circuit
(When testing, Nch transistor should be on.)

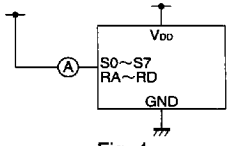


Fig. 4

OSC1 feedback current measurement circuit

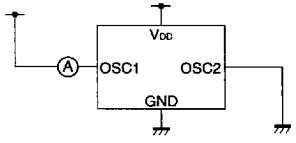


Fig. 5

Static current consumption measurement circuit
(Test in HALT mode, using HALT instruction
to enter mode.)

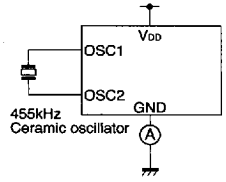


Fig. 6

Operating current consumption measurement circuit

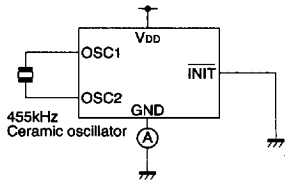


Fig. 7

Note: There are no RC or RD pins in the BU2461 and BU2462.

●Application example

This typical application example shows the BU2458 as the base chip in the remote control encoder shown below.

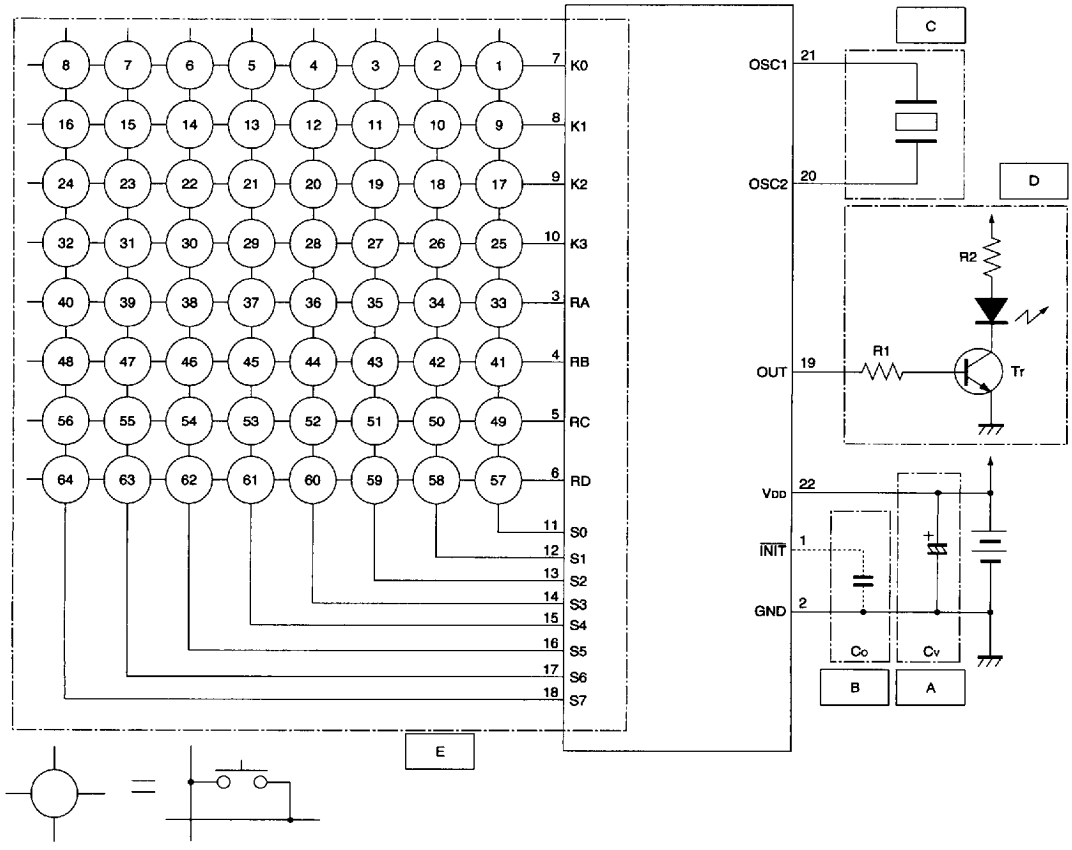


Fig. 8 Application circuit example (64-key remote control unit)

The following is a guide to peripheral circuits (A to E in the above diagram).

A : Power supply circuit
B : Initial reset circuit
C : Oscillation circuit
D : Infrared LED drive circuit
E : Key scan circuit

There are no RC or RD pins in the BU2461 and BU2462.

●Mask options

With the BU2456, BU2458, BU2459, BU2460, BU2461, and BU2462, the following options can be selected :

(1) The watchdog timer can be reset (or not reset) using the OUT output signal.
(2) The following input pins are equipped with a HALT cancel function : K0, K1, K2, K3, RA, RB, RC, RD.
(3) The following input / output pins are equipped with pull-up resistance : RA, RB, RC, RD.
(4) An internal capacitor can be provided (or not) for a ceramic oscillation circuit.
(5) Output states in HALT mode
S6 : LOW state (values prior to HALT state are retained)
S7 : LOW state (values prior to HALT state are retained)

●HALT function

The HALT state can be initiated by executing the HALT instruction. In the HALT state, the following apply :

(1) Oscillation stops, enabling an extremely low current consumption.
(2) The watchdog timer (WDT) is reset, and the S and OUT outputs go to LOW state.
(3) All values other than the WDT, S output and OUT output retain the values in effect prior to the HALT state.

However, the state of the S6 output and S7 output in the HALT mode can be selected using the mask options.

Ⓐ : Sets LOW state.
Ⓑ : The values in effect prior to the HALT state are maintained.

The HALT cancel function can be specified independently for each bit of the K inputs and R inputs, using the mask options.

If even one of the K inputs or R inputs has been set to LOW, the HALT state is cancelled.

If the HALT state is cancelled, the following occur :

1) The S output and OUT output values return to the values in effect prior to the HALT state.
2) In order to prevent operation before oscillation has stabilized, a wait timer is activated, and normal operation is then initiated after 1024 × 6 effective clock pulses.
3) At the point where normal operation is resumed, the WDT begins to count again from 0.

For K inputs and R inputs equipped with the HALT cancel function, if a LOW state has been input, the HALT instruction has the same effect as the NOP instruction.

●Description of the block diagram

(1) Program memory (ROM)

An internal ROM of 1024 words (64 words × 16 pages × 8 bits) for application software programs can be provided.

The scope of the program memory consists of the program counter PC ($A_0 \sim A_5$) and the page address register PA ($A_6 \sim A_9$), which are used to specify the address of the program register containing the next instruction (8 bits) to be executed.

The program memory is configured with one page consisting of 64 words, so that a maximum of 64 instructions can be stored on one page.

With evaluation chips, the ROM is attached externally, which makes it possible to carry out debugging and simulations of the pertinent system program. The program memory configuration is shown in Figure 9.

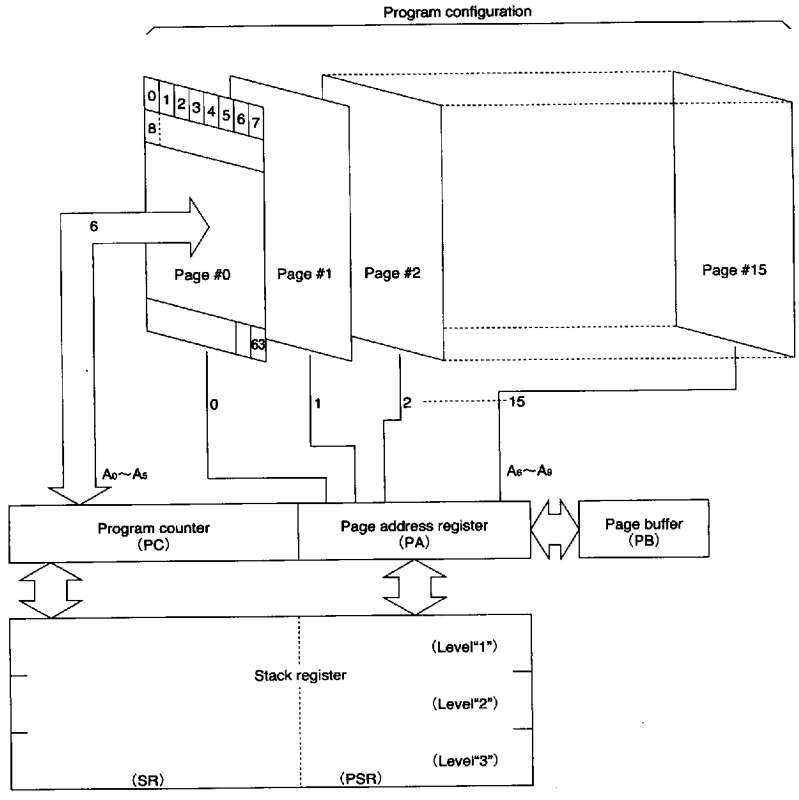


Fig. 9 Program memory configuration

(2) ROM address registers

The following registers are available to specify ROM addresses.

1) Page address register (PA)
   This contains the page numbers for the specified address in the ROM. The contents of the PA (4 bits) are decoded by a page decoder as one of 16 address lines.
2) Page buffer register (PB)
   New addresses are loaded into the PB, and are then shifted to the PA when the correct branch (BR) and subroutine call (CAL) are executed. The LPC instruction is used to load the PB.
3) Program counter (PC)
   This is used for word addresses on ROM pages. One instruction is selected by the page decoder from among the contents of the PC (6 bits).
4) Stack register (SR)
   This stores return word addresses in the Call Subroutine mode.

1. Page address register and page buffer register

The range of 0 ~ 15 pages in the ROM is defined by latching the four bits specifying the ROM page address. The page address register differs from the program counter in that normally it does not change. As long as the Page Change instruction is not executed, the program continues to remain on the same page.

To change the page address requires a two-step operation : ① write the page to which the program execution is to jump in the page buffer (execute the LPC instruction), and ② execute the BR or CAL instruction.

This is necessary because the instruction code consists of 8 bits, and the page and word cannot be specified at the same time.

If the return instruction (RTN) from a subroutine in a subroutine called from another page is executed, the page address is changed at the same time.

2. Program counter

The program counter is a 6-bit binary counter which is incremented each time an instruction is fetched, and specifies the address on the current page of the ROM containing the instruction which has to be executed next. To facilitate programming, the program counter is reset to a location of zero each time the power supply is turned on, and the page address is set to 0. Next, the program counter specifies the next ROM address, using a random sequence. When the BR, CAL, and RTN instructions are decoded, the switches are turned off between the various stages, and the address is not updated. Instead, for the BR and CAL instructions, the address data is read from the instruction operand (I_2 ~ I_7), and for the RTN instruction, the address is read from the stack register (Level 1).

3. Stack register

The stack register, which retains the return address when a subroutine is executed, is equipped with three levels for both the program counter (6 bits) and the page address register (4 bits). This enables subroutine nesting at three levels.

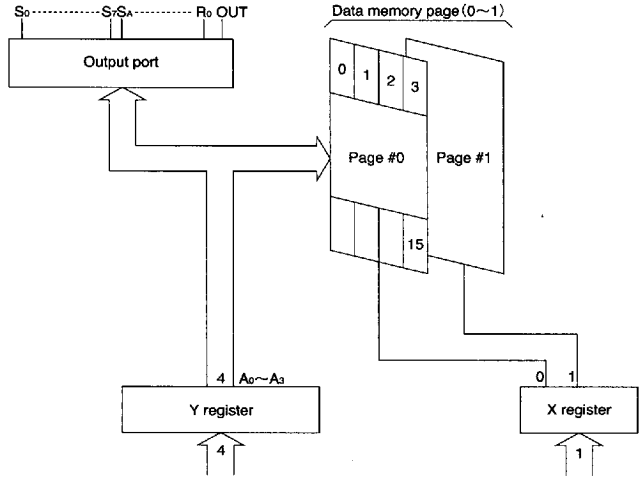


Fig. 10 Data memory configuration

(3) Data memory (RAM)
An internal RAM enables up to 32 words (16 words × 2 pages × 4 bits) to be stored.
The scope of the entire data memory is specified indirectly by the data pointers (X and Y), with the page number being specified by the one bit of the X register and the word on the page being specified by the four bits of the Y register. The data memory is configured of 16 words per page. This configuration is shown in Figure 10.

(4) X register (X)
The X register consists of one bit. This register specifies the page address in the data memory.

(5) Y register (Y)
The Y register consists of four bits. It acts as a data pointer or a general-purpose register.
The Y register specifies the address ($A_0 \sim A_3$) on the page in the data memory, and at the same time is used to specify the output port. Furthermore, it is used to specify the mode of the carrier signal output from the OUT output port. It can also be handled as a general-purpose register in the program.

(6) Accumulator (Acc)
This is a 4-bit register used when carrying out calculations. Data and data resulting from calculations are stored in this register.

(7) Arithmetic logic unit (ALU)
The arithmetic logic unit is configured of the addition and comparison units, which are connected in a 4-bit series, and a status logic (flag).
1) Calculation circuit (ALU)
The basic function of the addition/comparison unit is to carry out all addition and comparison operations. Subtraction is handled by reversing the Acc output [$\overline{\text{Acc}}$ + 1] and creating a complement.
2) Status logic
The status logic creates the ST, which is the flag that controls the program flow. If the calculation overflow and the two inputs are not equivalent, this flag is issued when the specified instruction is executed.

(8) Instruction decoder
Of the 43 types of instructions provided in the BU2458 series, 31 types are converted by the instruction decoding PLA into 16 types of micro-instructions, and when an instruction is executed, the instruction decoder handles operations such as connecting the data buses. Nine of the remaining 12 types of instructions, excluding the BR, CAL, and RTN instructions, are decoded by a decoder configured of only an AND matrix. This method works because the execution of these types of instructions does not require a large number of micro-instructions, and they operate in fairly isolated locations where they are not involved with other instructions. If the instruction is the BR, CAL, or RTN instruction, the following instruction is invalid. Therefore, these three instructions are not decoded using a decoder, but are decoded and executed directly by the hardware logic as the instruction code is read from the ROM. This avoids subsequent instructions being invalidated, so that the jump in program execution can be made to the pertinent address.

(9) Input/output circuits
The $K_0 \sim K_3$ ports are 4-bit input ports, and are pulled up by an MOS Tr resistance.
The $S_0 \sim S_7$ ports are output ports which can be set and reset independently. The output configuration is an Nch open drain circuit.
The $R_A \sim R_D$ ports are 4-bit input/output ports for which the outputs can be set and reset independently. The output configuration is an Nch open drain circuit. The input format is pulled up by an MOS Tr resistance (a using mask option). The OUT port can be used to drive large currents, and is a CMOS circuit (large current on the HIGH side).

(10) State counter (SC)
Figure 11 shows the basic machine cycle timing. All instructions consist of one byte, with the same execution time. Execution of one instruction consists of six clock pulses for the reading (Fetch) cycle and six clock pulses for the execution (Execute) cycle, for a total of 12 pulses, but in actuality, the two cycles overlap and are executed at the same time, so that it appears that the cycle execution (one machine cycle) is being completed in six clock pulses.
The exceptions to the execution time are the BR, CAL, and RTN instructions. With these three instructions, the subsequent command is invalidated because of the time required to change the address sequence. Therefore, the instruction is read in advance, so that execution can be completed within the fetch cycle.

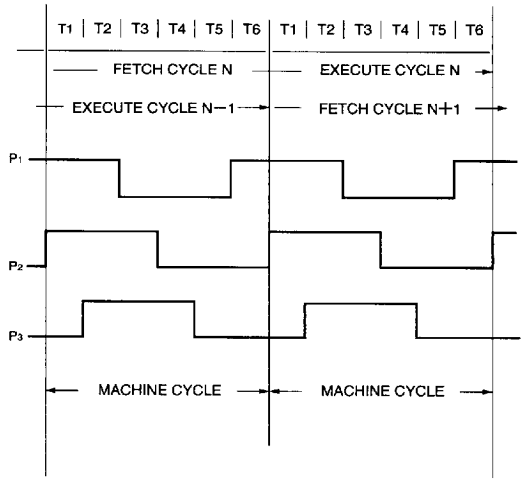


Fig. 11 Basic timing chart

(11) Clock generator

The BU2458 series has an internal clock generator. The oscillation circuit can be configured by attaching an external ceramic resonator (if an optional internal oscillation capacitor is used).

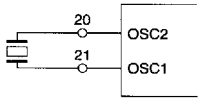


Fig. 12

*Depending on the resonator, an external capacitor may be necessary. Please refer to the recommended values of the pertinent manufacturer.)

(12) Reset function

Two reset functions are available to initialize the CPU. When the CPU is initialized, the program counter executes instructions from 0 page, 0 address.

1) Reset through an external pin

A reset can be executed by setting the $\overline{\text{INIT}}$ pin to LOW, which is done by specifying an interval of four or more machine cycles.

(One machine cycle = $1/f_{OSC} \times 6$)

If a reset is executed by turning on the power supply, a capacitor can be connected between the INIT pin and the GND and an integrated circuit configured with the internal pull-up resistance (approximately 400kΩ). This generates a reset pulse when the power supply is turned on. However, this method is effective only when a valid reset pulse is input while the power supply voltage is within the operating voltage range and the clock oscillation has stabilized.

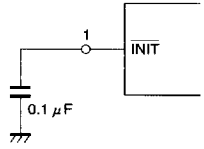


Fig. 13

*The capacitor value may need to be changed depending on the rise time of the power supply (this circuit example shows a power supply rise time of 10ms or less).

2) Reset using the internal Power On Reset circuit
The Power On Reset circuit is valid when the following conditions are satisfied. In this case, the external capacitor on the $\overline{INIT}$ pin may be omitted.

Table : Internal Power On Reset Standards
(V_{DD} = 3V, f_{CK} = 455kHz)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Conditions
Power supply rise time	t_{r1}	—	—	1	ms	VM_{in}=1V
Rise time when interruputed	t_{r2}	—	—	1	ms	
Power supply interrupt time	t_w	5	—	—	ms	

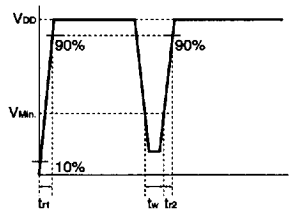


Fig. 14

(13) Carrier generator
The frequency and duty for the carrier signal output from the OUT output port are selected from among those listed in the table below, based on the value of the CMR (Carrier Mode Register) specified by the program.

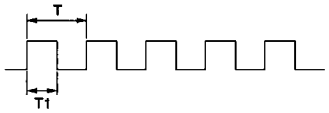


Fig. 15

CMR	OUT output signa
0	$T=1/f_{CAR}=12/f_{OSC}$ $T1/T=1/2$
1	$T=1/f_{CAR}=12/f_{OSC}$ $T1/T=1/3$
2	$T=1/f_{CAR}=8/f_{OSC}$ $T1/T=1/2$
3	$T=1/f_{CAR}=8/f_{OSC}$ $T1/T=1/4$
4	$T=1/f_{CAR}=11/f_{OSC}$ $T1/T=4/11$
5	No carrier output (same operation as $S_0 \sim S_7$)

* CMR is "0" on reset.

(14) Watchdog timer

The watchdog timer is configured of a 14-bit binary counter, and a clock signal of $f_{OSC}/6$ is input as the initial stage input. When this counter overflows, an internal reset signal is issued automatically, and the internal circuits are initialized. The detection time is set to $6 \times 2^{12}/f_{OSC}$ (102.4ms when f_{OSC} = 480kHz). Normally, this counter has to be reset prior to the overflow of the binary counter, and in addition to the WDTR instruction, a mask option makes it possible to use the OUT output signal to do this (when Y - reg = 8 and the SEO instruction is executed). A reset is carried out forcibly in the HALT state, and the counter resumes when the HALT state has been cancelled.

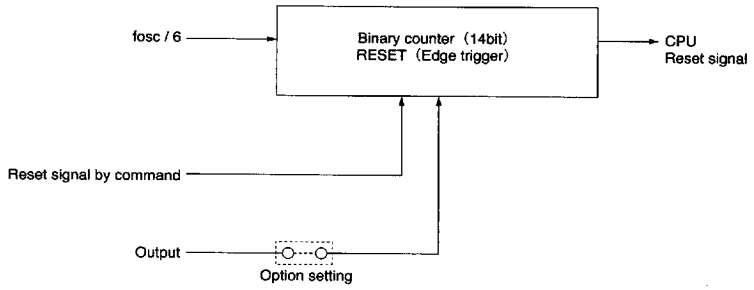


Fig. 16

●Instruction systems

(1) Abbreviations

Rohm's BU2458 series of 4-bit single-chip microcomputers has 43 types of basic instructions. These instructions are configured of the mnemonic abbreviations listed below.

1) Instructions related to operation

L—	:	Load
X—	:	Exchange
SE—	:	Set
RE—	:	Reset
D—	:	Decrement
I—	:	Increment
CL—	:	Clear
NEG—	:	Negate
A—	:	Add
S—	:	Subtract
BR	:	Branch
CAL	:	Call
RTN	:	Return
T—	:	Test
E—	:	Exclusive

2) Instructions related to constants

C [I (C)] : Immediate

Z : Zero

3) Instructions related to status

The status is set to "1" under the following conditions.

aNEb : $a \neq b$

aLEb : $a \leqq b$

—C : Carryout on A or I

—N : Not Borrow Out on S or D

—Z : Zero on NEG

TM : 1

4) Instructions relted to architecture block

P : Page Buffer Register

A : Accumulator Register

Y : Y-Register

M : RAM

$K_0 \sim K_3$: Input

$S_0 \sim S_7$, OUT : Output

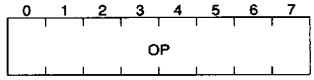
$R_A \sim R_D$: Input/output

(2) Instruction formats

All instructions are composed of 8-bits, including 2 fields : an operation code and an operand. Formats are classified by the type of operand.

1) Format I

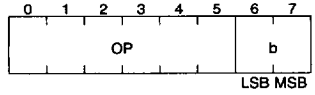
This format has no operand; all of the 8-bits are operation codes.



2) Format II

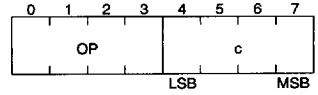
This format consists of 2 bits of operands and 6-bits of operation codes.

The 2-bits of operands are used to specify the RAM bit and the X register (Bits 1 and 7 are fixed at "0").



3) Format III

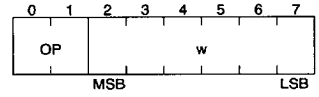
This format consists of 4-bits of operands and 4-bits of operation codes.The 4-bits of operands are used to specify the constant loaded into the RAM bit or the Y register, or to specify the comparison values for the comparison instruction. They may also be used to specify the ROM page address.



4) Format IV

This format consists of 6-bits of operands and 2-bits of operation codes.

The 2-bits of operands are used to specify the ROM address.



●Command functions

The BU2458 series is equipped with the 43 types of general commands shown below.

	Class.	Mnemonic	Function	ST*1	Instruction code	
1	Transfer from one register to another	LYA	Y←Acc	S	0010	0100
2		LAY	Acc←Y	S	0010	0011
3		CLA	Acc←0	S	0010	1111
4	Transfer from register to RAM	LMA	M(X, Y)←Acc	S	0000	0011
5		LMAIY	M(X, Y)←Acc, Y←Y+1	S	0010	0000
6	Transfer from RAM to register	LYM	Y←M(X, Y)	S	0010	0010
7		LAM	Acc←M(X, Y)	S	0010	0001
8		XMA	Acc↔M(X, Y)	S	0010	1110
9	Set constant	LYC	Y←I (c)	S	0100	(c)
10		LMCIY	M(X, Y)←I (c), Y←Y+1	S	0110	(c)
11		LXC	X←I (b)	S	0011	11(b)
12	RAM bit operation	SEM	M(X, Y, b)←1	S	0011	00(b)
13		REM	M(X, Y, b)←0	S	0011	01(b)
14		TM	When M (X, Y, b) = 1, set status	E	0011	10(b)
15	ROM address specification	BR	When ST = 1, branch to specified address	S	10 (w)	
16		CAL	When ST = 1, call subroutine of specified address	S	11 (w)	
17		RTN	Return from subroutine to main routine	S	0000	1111
18		LPC	PB←I (c)	S	0001	(c)
19	Calculation	AMAAC	Acc←Acc+M(X, Y)	C	0010	0101
20		SAMAN	Acc←M(X, Y)−Acc	B	0010	0111
21		IMAC	Acc←M(X, Y)+1	C	0010	1000
22		DMAN	Acc←M(X, Y)−1	B	0010	1010
23		IA	Acc←Acc+1	S	0000	1110
24		IYC	Y←Y+1	C	0010	1011
25		DAN	Acc←Acc−1	B	0000	0111
26		DYN	Y←Y−1	B	0010	1100
27		EMAA	Acc←M(X, Y)⊕Acc	S	0000	0001
28		NEGAZ	Acc←$\overline{\text{Acc}}$+1	Z	0010	1101
29	Comparison	ALEM	When Acc ≦ M (X, Y), set status	E	0010	1001
30		ALEC	When Acc ≦ 1 (c), set status	E	0111	(c)
31		MNEZ	If M (X, Y), is not equal to 0, set status	N	0010	0110
32		YNEA	If Y is not equal to Acc, set status	N	0000	0010
33		YNEC	If Y is not equal to I (c), set status	N	0101	(c)
34	Input	KNEZ	If K0 ~ K3 is not equal to 0, set status	N	0000	1001
35		LAK	Acc←K0~K3	S	0000	1000
36		RNEZ*3	If RA ~ RD is not equal to 0, set status	N	0000	1011
37		LAR	Acc←RA~RD	S	0000	1010
38	Output	SEO	Output (Y) ←1*2	S	0000	1101
39		REO	Output (Y) ←0*2	S	0000	1100
40	Other	WDTR	Watch Dog Timer Reset	S	0000	0101
41		HALT	Halt operation	S	0000	0110
42		CMS	CMR←Y	S	0000	0100
43		NOP	No operation	S	0000	0000

*1 ST : Indicates a condition for which the status (ST) changes, with the respective meanings noted below.
S : The status is set unconditionally when the instruction is executed.
C : The status is set if a carry or borrow condition occurs in the calculation results.
B : The status is set if a borrow condition does not occur in the calculation results.
E : The status is set if the result of a comparison is true.
N : The status is set if the result of a comparison is false.
Z : The status is set if the result of the calculation is zero.

*2 The following are carried out based on the contents of the Y register.

Y register value	Operation
0~7	SEO instruction: S (Y)←1 REO instruction: S (Y)←0
8	SEO instruction: OUT (CMR)←1 The OUT output pin alternates repeatedly between HIGH and LOW, at the carrier frequency.(If CMR = 5, the status is fixed at HIGH.) REO instruction: OUT (CMR)←0
9	SEO instruction: S0~ S7←1 (High-Z) REO instruction: S0~ S7←0
A~D	SEO instruction: R (Y)←1 REO instruction: R (Y)←0
E	SEO instruction: RA~ RD←1 REO instruction: RA~ RD←0
F	SEO instruction: S0~ S7, RA ~ RD←1 REO instruction: S0~ S7, RA ~ RD←0

* The BU2461 and BU2462 do not have the RNEZ command.

●Notes about the instruction system

The following describes each of the 43 types of basic instructions in the BU2458 series, in greater detail.

(1) Notation format

Descriptions are noted using the mnemonic abbreviations listed in the "Instruction Functions" table at the beginning of the instruction. The basic items noted below are indicated to facilitate understanding, with additional notes concerning functions and other information.

1) Notation format

Item

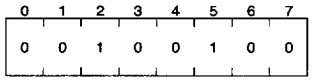
①Naming : Full name of mnemonic
②Binary operation code : Displayed in diagram
③Status : Check of status functions
④Format : Type of instruction format (I to IV)
⑤Operand : Omitted for Format I instructions
⑥Function : Description of function

(2) Descriptions of instructions

1) LYA

Naming : Load Y - Register from Accumulator

Binary operating code :



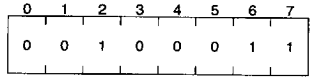
Status : Set

Format : I

Function : Y ← Acc

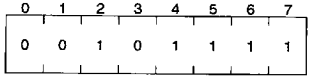
〈Description〉 The contents of the accumulator are transferred unconditionally to the Y register. The contents of the accumulator do not change.

2) LAY
Naming : Load Accumulator from Y - Register
Binary operating code :



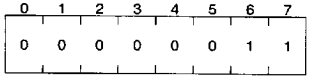
Status : Set
Format : I
Function : A ← Ycc
〈Description〉 The four bits of the Y register are transferred unconditionally to the accumulator. The contents of the Y register do not change.

3) CLA
Naming : Clear Accumulator
Binary operating code :



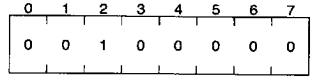
Status : Set
Format : I
Function : Acc ← 0
〈Description〉 The contents of the accumulator are set unconditionally to 0.

4) LMA
Naming : Load Memory from Accumulator
Binary operating code :



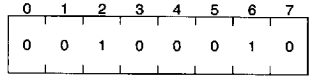
Status : Set
Format : I
Function : M (X, Y) ← Acc
〈Description〉 The 4-bit contents of the accumulator are accumulated in the memory (RAM) location, in the addresses specified by the X and Y registers. The contents of the accumulator do not change.

5) LMAIY
Naming : Load Memory from Accumulator and Increment Y-Register
Binary operating code :



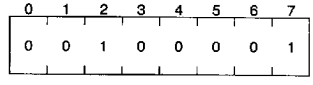
Status : Set
Format : I
Function : M (X, Y) ← Acc Y ← Y + 1
〈Objective〉 The Y register continuously addresses the pages of the 16 RAM words, and the addressed word is set as the accumulator value.
〈Description〉 The contents of the accumulator are accumulated in the RAM location addressed by the X and Y registers.
Next, the contents of the Y register are incremented by one. The contents of the accumulator do not change.

6) LYM
Naming : Load Y-Register from Memory
Binary operating code :



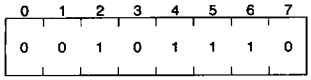
Status : Set
Format : I
Function : Y ← M (X, Y)
〈Description〉 The contents of the RAM location addressed by the X and Y registers are loaded to the register. The contents of the memory do not change.

7) LAM
Naming : Load Accumulator from Memory
Binary operating code :



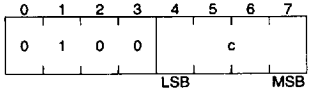
Status : Set
Format : I
Function : Acc ← M (X, Y)
〈Description〉 The contents of the RAM location addressed by the X and Y registers are loaded to the accumulator. The contents of the memory do not change.

8) XMA
Naming : Exchange Memory and Accumulator
Binary operating code :



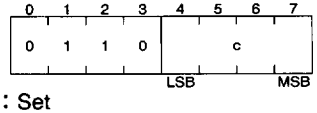
Status : Set
Format : I
Function : M (X, Y) ⇆ Acc
〈Description〉 The contents of the memory addressed by the X and Y registers are exchanged with the contents of the accumulator. For example, in order to carry out a calculation, this instruction can be used to read the memory words in the accumulator and save the current accumulator contents to the RAM. The contents can then be returned to the accumulator at the next XMA instruction.

9) LYC
Naming : Load Y-Register from Immediate
Binary operating code :



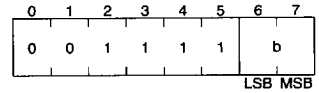
Status : Set
Format : III
Operand : Constant 0 ≦ I (c) ≦ 15
Function : Y ← I (c)
〈Objective〉 This loads a constant to a register. In an ordinary usage example, Y is set in a specified RAM word address, an address for a selected output line is specified, or a carrier signal to output from the OUT output port is specified, by setting Y. It can also be used to initialize the Y register for loop control.
〈Description〉 A 4-bit value is transferred from the c field of the instruction to the Y register.

10) LYCIY
Naming : Load Memory from Immediate and Increment Y-Register
Binary operating code :



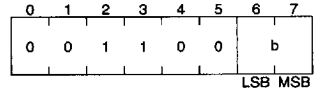
Status : Set
Format : III
Operand : Constant 0 ≦ I (c) ≦ 15
Function : M (X, Y) ← I (c), Y ← Y + 1
〈Description〉 A 4-bit value is transferred from the c field of the instrtion and accumulated in the RAM location addressed by the X and Y registers. Next, the contents of the Y register are incremented by 1.

11) LXC
Naming : Load X-Register from Immediate
Binary operating code :



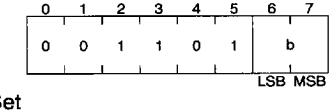
Status : Set
Format : II
Operand : X field address 0 ≦ I (b) ≦ 1
Function : X ← I (b)
〈Description〉 The constant value is loaded to the X register. This is used to set the index of the desired RAM page in the register. The b field of Bit 1 of the instruction is loaded to the X register.

12) SEM
Naming : Set Memory Bit
Binary operating code :



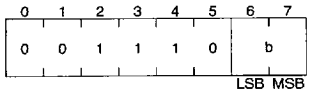
Status : Set
Format : II
Operand : Bit address 0 ≦ I (b) ≦ 3
Function : M (X, Y, b) ← 1
〈Description〉 In accordance with the selection of the b field of the operand, the first bit of the four bits is set in the addressed RAM memory as a logic of 1, depending on the contents of the X and Y registers.

13) REM
Naming : Reset Memory Bit
Binary operating code :



Status : Set
Format : II
Operand : Bit address 0 ≦ I (b) ≦ 3
Function : M (X, Y, b) ← 0
〈Description〉 In accordance with the selection of the b field of the instruction, the first bit of the four bits is set in the addressed RAM memory as a logic of 0, depending on the contents of the X and Y registers.

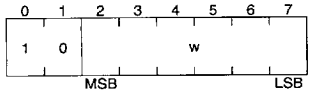
14) TM
Naming : Test Memory Bit
Binary operating code :



Status : To status of comparison results
Format : II
Operand : Bit address 0 ≦ I (b) ≦ 3
Function : M (X, Y, b) ← 1 ?
ST ← 1 (when M (X, Y, b) = 1)
ST ← 0 (when M (X, Y, b) = 0)

〈Objective〉 This test to see whether the logic of the selected memory bit is 1, and sets the status based on the result of the test.

15) BR
Naming : Branch on status 1
Binary operating code :



Status : Sets the status after executing the instruction, conditionally, depending on the status
Format : IV
Operand : Branch address (W)
Function :
If ST = 1, PA ← PB, PC ← I (W)
If ST = 0, PC ← PC + 1, ST ← 1

Note: In actuality, PC is a pseudo-random counter which specifies the next address in a fixed sequence.

〈Objective〉 Depending on the program, this can be used to change the order in which instructions are executed in a normally sequential program. The branch is conditional, based on the status of the results of executing the instruction.

〈Description〉

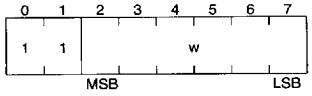
●This branching instruction is normally conditional, based on the status.

①If the status is "Reset" (the logic is 0), the next sequential instruction is executed, rather than the program being executed in the correct sequential order.

②If the status is "Set" (the logic is "1"), the program is executed based on the following types of action.

●There are two types of branching, long and short. Short branching is used if the address is on the current page, and long branching is used if the address is on a different RAM page. The type of branching carried out is determined by the contents of the PB register. In order to execute long branching, the contents of the PB register must be modified with the desired page address, and this is done using the Load PB Register (LPC) instruction.

16) CAL
Naming : Subroutine Call on status 1
Binary operating code :



Status : Sets the status after executing the instruction, conditionally, depending on the status
Format : IV
Operand : Subroutine code less I (W)
Function :

If ST = 1,	PC ← I (W)	PA ← PB
	SR1 ← PC + 1	PSR1 ← PA
	SR2 ← SR1	PSR2 ← PSR1
	SR3 ← SR2	PSR3 ← PSR2
If ST = 0,	PC ← PC + 1	PB ← PA, ST ← 1

Note: In actuality, PC has a pseudo-random counter in relation to the next instruction.

〈Description〉

●In the program, control can be shifted between common subroutines. The loaded instruction retains the return address, so subroutines can be called out from various locations throughout the program, and the Call Return instruction (RTN) can be used in the subroutine to return execution correctly to the loaded address.Loading of addresses and instructions is normally carried out conditionally, based on the status.

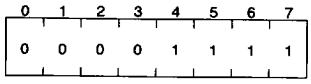
①If the status is "Reset", loading does not take place.

②If the status is "Set", loading is carried out correctly.

Because the subroutine stack (SR) has three levels, subroutines can be handled on up to three levels. Long subroutine calls (those in which another page is called out) can also be executed from any level.

●Long subroutine calls (those in which another page is called out) are enabled by executing the LPC instruction prior to CAL. Omitting the LPC instruction (and PA = PB) serves as a short subroutine call (calling a subroutine on the same page).

17) RTN
Naming : Return from Subroutine
Binary operating code :

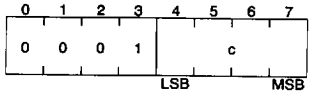


Status : Set
Format : I
Function : PC ← SR1 PA, PB ← PSR1
SR1 ← SR2 PSR1 ← PSR2
SR2 ← SR3 PSR2 ← PSR3
SR3 ← SR3 PSR3 ← PSR3
ST ← 1

〈Objective〉 This returns execution from the called-out subroutine to the program from which control was called.

〈Description〉 Information about the return address taken from the stack register (SR1) is sent to PC, and execution returns to the main routine. At the same time, the contents of the page stack register (PSR1) are sent to PA and PB.

18) LPC
Naming : Load Page Buffer Register from Immediate
Binary operating code :

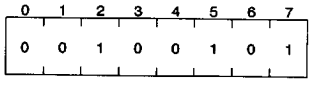


Status : Set
Format : III
Operand : ROM page address 0 ≦ I (b) ≦ 15
Function : PB ← I (c)

〈Objective〉 This loads a new ROM page address to the page buffer (PB) register. This operation is necessary when an instruction is used to call a long branch.

〈Description〉 The PB register is loaded in accordance with the four bits from the c field of the instruction.

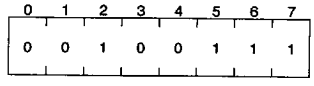
19) AMAAC
Naming : Add Accumulator to Memory and Status 1 on Carry
Binary operating code :



Status : Carry to status
Format : I
Functions : Acc ← M (X, Y) + Acc
ST ← 1 (when total > 15)
ST ← 0 (when total ≦ 15)

〈Description〉 The ontents of the memory location addressed by the X and Y registers are added to those of the accumulator, and the results are accumulated in the accumulator. The carry information is then sent to the status as the result. If the total is larger than 15, a carry operation occurs, and "1" is set as the status. The contents of the memory do not change.

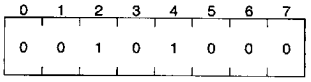
20) SAMAN
Naming : Subtract Accumulator to Memory and Status 1 not Borrow
Binary operating code :



Status : Carry to status
Format : I
Functions :
Acc ← M (X, Y) - Acc
ST ← 1 (when A ≦ M (X, Y))
ST ← 0 (when A > M (X, Y)) } Initial conditions

〈Description〉 Two's complement addition is carried out on the contents of the accumulator, and the contents of the accumulator subtracted from the memory word addressed by the X and Y registers, with the results being accumulated in the accumulator. If the value in the accumulator is smaller than the memory word, or is equivalent to it, the status is "Set", to indicate that a borrow operation was not carried out. If the value in the accumulator is larger than the memory word, a borrow operation occurs, and the status is reset to 0.

21) IMAC
Naming : Increment Memory and Status 1 on Carry
Binary operating code :

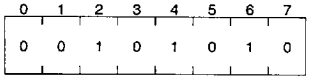


Status : Carry to status
Format : I
Functions :

Acc ← M (X, Y) + 1
ST ← 1 (when M (X, Y) ≧ 15)
ST ← 0 (when M (X, Y) < 15)
} Initial conditions

〈Description〉 The contents of the memory addressed by the X and Y registers are read out. "1" is added to this word, and the result are accumulated in the accumulator. The carry information is sent to the status as the result. If the total is greater than 15, the status is "Set". The memory does not change.

22) DMAN
Naming : Decrement Memory and Status 1 on Not Borrow
Binary operating code :

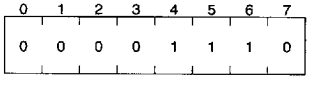


Status : Carry to status
Format : I
Functions :

Acc ← M (X, Y) - 1
ST ← 1 (when M (X, Y) ≧ 1)
ST ← 0 (when M (X, Y) = 0)
} Initial conditions

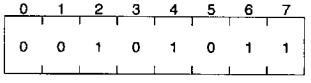
〈Description〉 The contents of the memory addressed by the X and Y registers are red out. "1" is subtracted from this word (F_H addition), and the results are accumulated in the accumulator. The carry information is sent to the status as the result. If the memory is greater than 1, the status is "Set", to indicate that a borrow operation was not carried out. The contents of the memory do not change.

23) IA
Naming : Increment Accumulator
Binary operating code :



Status : Set
Format : I
Functions : Acc ← Acc + 1
〈Description〉 "1" is added to the contents of the accumulator, and the result is returned to the accumulator. There is no influence on the carry status.

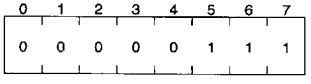
24) IYC
Naming : Increment Y-Register and 1 on Carry
Binary operating code :



Status : Carry to status
Format : I
Functions : Y ← Y + 1
ST ← 1 (when Y ≧ 15)
ST ← 0 (when Y < 15)
} Initial conditions

〈Description〉 "1" is added to the contents of the Y register, and the result is returned to the Y register. The carry information is sent to the status as the result. If the total is greater than 15, the status is "Set".

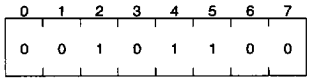
25) DAN
Naming : Decrement Accumulator and Status 1 Borrow
Binary operating code :



Status : Carry to status
Format : I
Functions : Acc ← Acc - 1
ST ← 1 (when A ≧ 1)
ST ← 0 (when A = 0)
} Initial conditions

〈Descrition〉 "1" is subtracted from the contents of the accumulator (F_H addition). If a borrow operation is carried out, the status is reset to a logic of 0. If the contents of the accumulator are greater than 1, no borrow operation is carried out, and the status is set to 1.

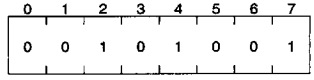
26) DYN
Naming : Decrement Y-Register and Status 1 on Not Borrow
Binary operating code :



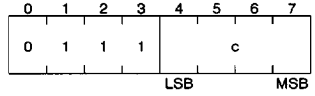
Status : Carry to status
Format : I
Functions : Y ← Y - 1
ST ← 1 (when Y ≧ 1)
ST ← 0 (Y = 0)
} Initial conditions
〈Objective〉 Thissubtracts only "1" from the contents of the Y register.
〈Description〉 "1" is subtracted from the contents of the Y register. This is done by adding a negative 1 (F_H). The carry information is sent to the status as the result. If the result is not equal to 15, the status is reset to 0, to indicate that a borrow operation was carried out.
27) EMAA
Naming : Exclusive OR Memory and Accumulator
Binary operating code :

Status : Set
Format : I
Function : Acc ← M (X, Y) ⊕ Acc
〈Description〉 An exclusive logical OR is carried out on the contents of the memory addressed by the X and Y registers and the contents of the accumulator, and the results are accumulated in the accumulator.
28) NEGAZ
Naming : Negative Accumulator and Status 1 on Zero
Binary operating code :

Status : Carry to status
Format : I
Functions :
Acc ← $\overline{Acc}$ + 1
ST ← 1 (when A = 0)
ST ← 0 (when A is not equal to 0)
} Initial conditions
〈Objective〉 This makes the word in the accumulator a two's complement.
〈Description〉 The two's complement of the accumulator is calculated by adding "1" to the one's complement of the accumulator, and the results are accumulated in the accumulator. The carry information is sent to the status. If the contents of the accumulator are 0, a carry operation occurs, and the status is set to 1.
29) ALEM
Naming : Accumulator Less Equal Memory
Binary operating code :



Status : Carry to status
Format : I
Functions : Acc ≦ M (X, Y)
ST ← 1 (when Acc ≦ M (X, Y))
ST ← 0 (when Acc <= M (X, Y))
〈Description〉 The value in the accumulator is subtracted from the contents of the memory location addressed by the X and Y registers, using complement addition. The carry information is sent to the status as the result. A status of 1 indicates that the value in the accumulator is less than or equal to the memory word. Neither the contents of the memory nor those of the accumulator change.
30) ALEC
Naming : Accumulator Less Equal Immediate
Binary operating code :



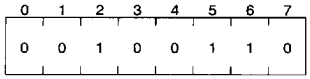
Status : Carry to status
Format : III
Operand : Constant value 0 ≦ I (c) ≦ 15
Functions : Acc ≦ I (c) ?
ST ← 1 (when Acc ≦ I (c))
ST ← 0 (when Acc > I (c))
〈Objective〉 This compares the contents of the accumulator and the constant value arithmetically.

〈Description〉 The value in the accumulator is subtracted from the constant (which is in the c field of the instruction), using complement addition. The carry information is sent to the status as the result. If the accumulator contents are less than or equal to the constant, the status is "Set". The contents of the accumulator do not change.

31) MNEZ

Naming : Memory Not Equal Zero

Binary operating code :



Status : Comparison result to status

Format : I

Functions : M (X, Y) ≠ 0 ?

ST ← 1 (when M (X, Y) ≠ 0)

ST ← 0 (when M (X, Y) = 0)

〈Objective〉 This compares the memory word to 0.

〈Description〉 The contents of the memory addressed by the X and Y registers are compared logically to 0. The comparison information is sent to the status as the result. If the memory value is not 0, the status is "Set".

32) YNEA

Naming : Y-Register Not Equal Accumulator

Binary operating code :

Status : Comparison result to status

Format : I

Functions : Y ≠ Acc ?

ST ← 1 (when Y ≠ Acc)

ST ← 0 (when Y = Acc)

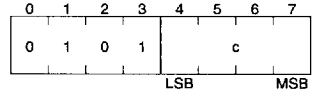
〈Objective〉 This compares the contents of the Y register to those of the accumulator, to see if they match.

〈Description〉 The contents of the Y register are compared logically to the contents of the accumulator. The comparison information is sent to the status as the result. If the contents do not match, the status is "Set".

33) YNEC

Naming : Y-Register Not Equal Immediate

Binary operating code :



Status : Comparison result to status

Format : III

Operand : Constant value 0 ≦ I (c) ≦ 15

Functions : Y ≠ I (c) ?

ST ← 1 (when Y ≠ I (c))

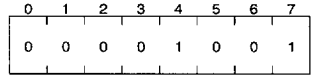
ST ← 0 (when Y = I (c))

〈Description〉 The constant in the Y register is compared logically to the four bits in the c field of the instruction. The comparison result is sent to the status. If the erand and the contents of the Y register do not match, the status is set to 1.

34) KNEZ

Naming : K Not Equal Zero

Binary operating code :



Status : Status set only if comparison results do not match

Format : I

Function : When $K_0 \sim K_3$ are not equal to 0, status is set

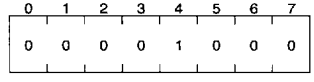
〈Objective〉 This tests to see whether or not $K_0 \sim K_3$ are 0.

〈Description〉 The data in $K_0 \sim K_3$ are compared to 0. The comparison information is sent to the status. If data which is not 0 has been input, the status is set.

35) LAK

Naming : Load Accumulator from K

Binary operating code :



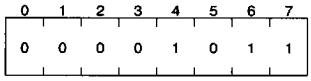
Status : Set

Format : I

Function : Acc ← $K_0 \sim K_3$

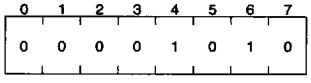
〈Description〉 The data in $K_0 \sim K_3$ are sent to the accumulator.

36) RNEZ
Naming : R Not Equal Zero
Binary operating code :



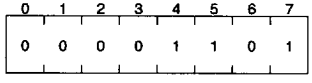
Status : Status set only if comparison results do not match
Format : I
Function : When $R_A \sim R_D$ are not equal to 0, status is set
〈Objective〉 This tests to see whether or not $R_A \sim R_D$ are 0.
〈Description〉 The data in $R_A \sim R_D$ are compared to 0. The comparison information is sent to the status. If data which is not 0 has been input, the status is set.

37) LAR
Naming : Load Accumulator from R
Binary operating code :



Status : Set
Format : I
Function : Acc ← $R_A \sim R_D$
〈Description〉 The data in $R_A \sim R_D$ are sent to the accumulator.

38) SEO
Naming : Set Output Register Latch
Binary operating code :



Status : Set
Format : I
Function :

S (Y) ← 1	$0 \leqq Y \leqq 7$
OUT ← 1 (CMR = 5)	Y = 8
$S_0 \sim S_7$ ← 1 (High-Z)	Y = 9
R (Y) ← 1	$A_H \leqq Y \leqq D_H$
RA ~ R_D ← 1	$Y = E_H$
$S_0 \sim S_7$, $R_A \sim R_D$ ← 1	$Y = F_H$

〈Objective〉
When $0 \leqq Y \leqq 7$, this sets one S output line to a logic of 1.
When Y = 8, the OUT output pin alternates between HIGH and LOW at the carrier frequency.
When Y = 9, this sets the $S_0 \sim S_7$ output line to a logic of 1.
When $A_H \leqq Y \leqq D_H$, this sets one R output line to a logic of 1.
When $Y = E_H$, this sets the $R_A \sim R_D$ output lines to a logic of 1.
When $Y = F_H$, this sets the $S_0 \sim S_7$ and $R_A \sim R_D$ output lines to a logic of 1.
〈Description〉 When the content of the Y register are between 0 and 7, the contents of the Y register select the appropriate S output ($S_0 \sim S_7$).
When the contents of the Y register are 8, the OUT output port is selected.
When the contents of the Y register are 9, $S_0 \sim S_7$ are all selected.
When the contents of the Y register are between $A_H \sim D_H$, the contents of the Y register select the appropriate R output ($R_A \sim R_D$).
When the contents of the Y register are E_H, $R_A \sim R_D$ are all selected.
When the contents of the Y register are F_H, $S_0 \sim S_7$ and $R_A \sim R_D$ are all selected.

39) REO
Naming : Reset Output Register Latch
Binary operating code :

Status : Set
Format : I
Function :

S (Y) ← 0	$0 \leqq Y \leqq 7$
OUT ← 0	Y = 8
$S_0 \sim S_7$ ← 0	Y = 9
R (Y) ← 0	$A_H \leqq Y \leqq D_H$
$R_A \sim R_D$ ← 0	$Y = E_H$
$S_0 \sim S_7$, $R_A \sim R_D$ ← 0	$Y = F_H$

〈Objective〉
When 0 ≦ Y ≦ 7, this sets one S output line to a logic of 0.
When Y = 8, the OUT output pin is set to 0.
When Y = 9, this sets the S0 ~ S7 output lines to a logic of 0.
When A_H ≦ Y ≦ DH, this sets one R outpine to a logic of 0.
When Y = E_H, this sets the R_A ~ R_D output lines to a logic of 0.
When Y = F_H, this sets the S_0 ~ S_7 and R_A ~ R_D output lines to a logic of 0.
〈Description〉 When the contents of the Y register are between 0 and 7, the contents of the Y register select the appropriate S output (S_0 ~ S_7).
When the contents of the Y register are 8, the OUT output port is selected.
When the contents of the Y register are 9, S_0 ~ S_7 are all selected.
When the contents of the Y register are between A_H ~ D_H, the contents of the Y register select the appropriate R output (R_A ~ R_D).
When the contents of the Y register are E_H, R_A ~ R_D are all selected.
When the contents of the Y register are F_H, S_0 ~ S_7 and R_A ~ R_D are all selected.

40) WDTR
Naming : Watch Dog Timer Reset
Binary operating code :

0	1	2	3	4	5	6	7
0	0	0	0	0	1	0	1

Status : Set
Format : I
Function : Resets the watchdog timer
〈Objective〉 Normally, the counter for the watchdog timer must be reset before it overflows, and this instruction is used to control the reset signal.

41) HALT
Naming : HALT
Binary operating code :

0	1	2	3	4	5	6	7
0	0	0	0	0	1	1	0

Status : Set
Format : I
Function : Implements the HALT function
〈Objective〉 This stops oscillation and sharply reduces the amount of current consumed.
(See Chapter 1, section 1-7, HALT Function.)

42) CMS
Naming : Carrier Mode Set
Binary operating code :

0	1	2	3	4	5	6	7
0	0	0	0	0	1	0	0

Status : Set
Format : I
Function : CMR ← Y
〈Description〉 This selects the carrier signal output from the OUT output port.

43) NOP
Naming : No Operation
Binary operating code :

0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0

Status : Set
Format : I
Function : No operation

●Development support systems

(1) Software support (RDS-BU2)

In order to facilitate software development with the BU2458 series, the following software configuration has been provided as a development support tool.

1) Software development support tools

The following software development support tools are available.

1. Personal computer system (RDS-BU2)

The personal computer system requires the ability to run MS-DOS * as the operating system (OS). A sample system configuration is shown for reference in Figure 17.

2) Software configuration

The following are available as software development support programs. These software programs require the ability to run MS-DOS * as the OS.

1. Text editor

This software program allows source programs to be created and modified. It can be used as a general-purpose screen editor.

2. Assembler

This is used to convert source programs written as mnemonics to objects, and to create assembly lists and object files. The general-purpose cross-assembler PROASM-II ** is used.

3. BUS2458 instruction library

This can be used by including the BU2458 macro library in the general-purpose cross-assembler PROASM-II **.

4. Simulator

Using objects which have been created, target machines can be run in simulated operation in the host machine, in order to monitor port and register statuses. This is used to run programs in the CPU in advance and to debug them, before running them with the evaluation board.

When object files confirmed with the above software programs have been completed, trial production of microcomputers for the pertinent system can be carried out at Rohm.

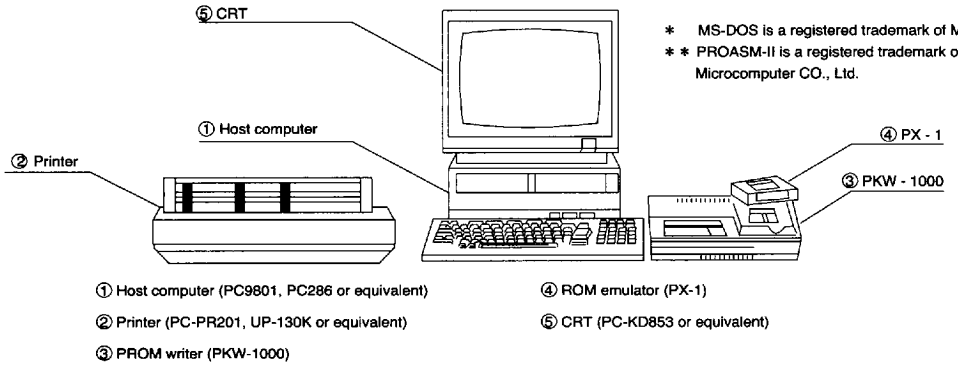


Fig. 17

(2) Development support system software

This section explains the various syntaxes used to activate this development support system, and the ways in which the various software programs are used.

1) Source program formats

Source programs are created using a screen editor in MS-DOS *. Source program statements consist of the following four fields.

Label field	Operation field	Operand field	Comment field

These fields must be lined up in the configuration shown above.

Blanks are used as delimiters between fields. Consequently, the next field begins with the first character following the blank. The first character in a label or operation field must be an English alphabetic character. Also, if the first column is blank, it indicates that the label field is not limited.

Figure 17 shows an example of a source program, and Figure 19 shows an example of the assembly output for this program.

1. Label field

The label field is designed to indicate the location of one instruction in the program in a way that is easy to reference and easy to remember, and is configured of between 1 and 128 English alphabetic letters. Label fields must be clearly differentiated, so that locations are also clearly differentiated.

2. Operation field

Operation fields are configured of operation codes which indicate the function which the instruction implements. There must be at least one blank between an operation field and a label field.

3. Operand field

Operand fields are configured of operands which indicate the target of the action being implemented. There must be at least one blank between an operation code and an operand.

4. Comment field

Comments begin with a semi-colon and end with a carriage return and a line field.

```
        INCLUDE BU2458. LIB
                ORG     $000
                LXC     0
                LYC     0
AAA             LMCIY   0
                YNEC    0
                BR      AAA
                END
```

Fig. 18 Example of source program

```
IA80 Ver.2.08(C) 1985 by IWASAKI / K.M.C PAGE 1
1 0000                  INCLUDE BU2458. LIB
2 0000                  list
3 0000                  ORG     $000
4 0000  3C  +           LXC     0
5 0001  40  +           LYC     0
6 0003  60  + AAA       LMCIY   0
7 0007  50  +           YNEC    0
8 000f  83  +           BR      AAA
9 001f                  END
No Fatal error(s)
```

Fig. 19 Example of assembly output

(3) Hardware support

The following hardware support systems are available, to make it faster and easier to develop target systems for the BU2458 series.

1) BU2494 (Evaluation chip for the BU2458 series)

1. Overview

The BU2494 is an IC designed for evaluation of the BU2458 series of 4-bit, single-chip microcomputers. Basically, it is configured of the same IC chips as the BU2458, but instead of the internal mask ROM unit used to write programs, it is configured so that a program memory (ROM or RAM) can be externally attached.

By writing programs to this externally attached program ROM, the user can treat the system in the same way as the ICs of the BU2458 series for the target system.

Therefore, incorporating the BU2494 and the EV2418B evaluation board with the externally attached program memory into user system makes program and system debugging easier.

2. Features

●Emulation of the BU2458 series of 4-bit, single-chip microcomputers is enabled.

●The system is configured of a CMOS processor which allows comprehensive evaluation of the BU2458 series.

●The instruction system is the same as that of the BU2458 series.

●External attachment of a program ROM (2732, 2764, 27128, or 27256) on the BU2494 enables the system to be handled just like the BU2458 series.

●Using the $PA_0 \sim PA_3$ and $PC_0 \sim PC_5$ which control up to 1024 words of ROM addresses (8 bits/word) and the ROM data outputs $I_0 \sim I_7$ enables linking with an external program ROM.

3. Block diagram

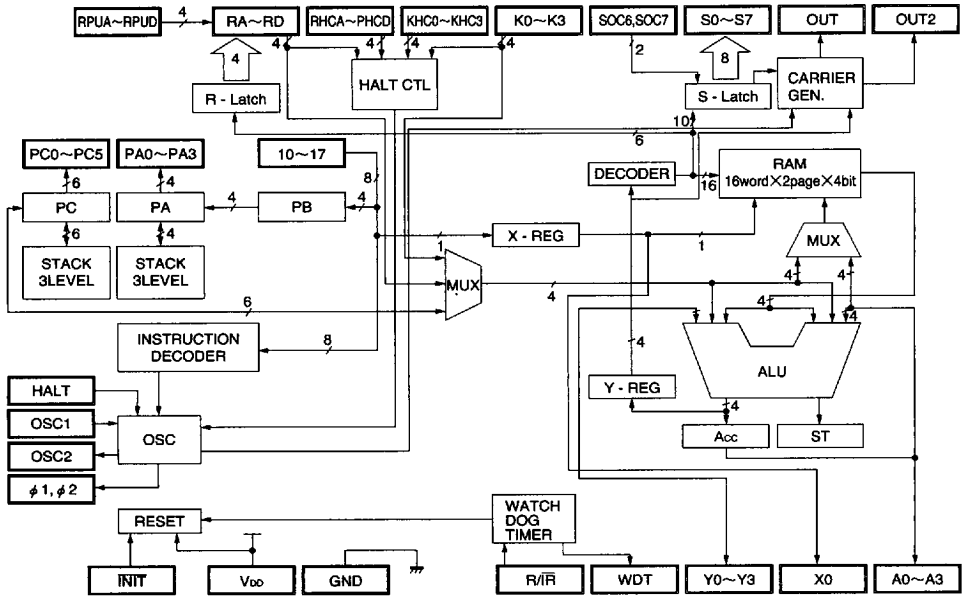


Fig. 20

4. Pin assignments (pin functions)

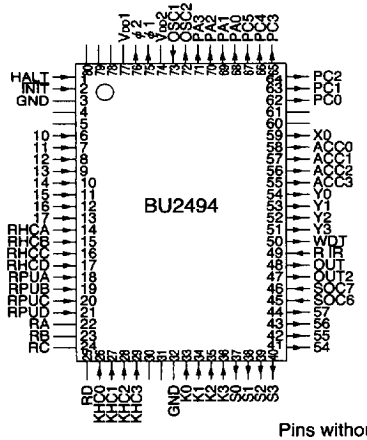


Fig. 21

Pins without names are not connected.

5. External dimensions (Unit: mm)

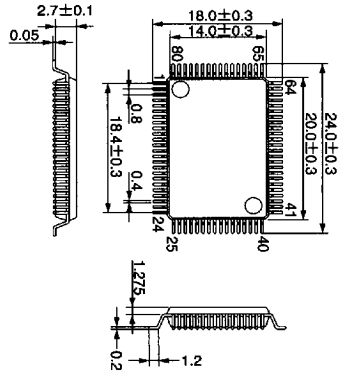


Fig. 22

6. Description of pin functions

Pin Name	I/O	Function
V_{DD1}		Used to connect 2.0 V ~ 4.0 V power supply.
GND		Reference voltage for all inputs and outputs (0 V).
$\overline{INIT}$	Input	Manual reset input. Setting this pin to LOW initializes the S output port and sets the ROM address to 0 page, 0 address.
K0~K3	Input	4-bit input ports. Internal pull-up resistance. Setting this pin to LOW cancels the HALT function. (The HALT cancel function can be set using the KHC pin.)
S0~S7	Output	Each of these can be set and reset independently (or all at once). The output format is Nch open drain.
OUT	Output	Remote control signal output port which can be used to drive large currents. The output format is CMOS output (output HIGH current side).
OSC1	Input	A ceramic resonator is connected between this pin and OSC2. There is internal feedback resistance between this pin and OSC2.
OSC2	Output	A ceramic resonator is connected between this pin and OSC1.
KHC0~3 RHCA~D	Input	The HALT cancel function is turned on and off by means of the K and R input to these pins. Setting these pins to HIGH makes the HALT cancel function effective based on the corresponding K and R input.
HALT	Input	External HALT input; however, HALT activation using an instruction takes precedence.
$\phi 1$, $\phi 2$	Output	Timing signal outputs
A0~A3	Output	These are used to output the accumulator contents.
Y0~Y3	Output	These are used to output the contents of the Y register.
V_{DD2}	—	Power supply for interface block with EPROM. A power supply of 4.5 ~ 5.5 is connected.
R/$\overline{IR}$	Input	Watchdog timer mode setting pins. Setting these to LOW initiates a reset of the timer using the OUT signal.
WDT	Output	Monitor output for the reset signal of the watchdog timer.
PC0~5, PA0~3	Output	These are used to output addresses to an external EPROM.
I0~I7	Input	These are used to input data from an EPROM.
RA~RD	Input/ output	These are 4-bit input/output ports. Internal pull-up resistance (set via the RPU input). Setting these pins to LOW cancels the HALT function. (The HALT cancel function can be set using the RHC pin.)
SOC6, SOC7	Input	Port status setting pins for S6 and S7 in HALT mode. Setting these pins to HIGH outputs a LOW state when in HALT mode.
OUT2	Output	Signal output pin for signal output from the OUT output port, excluding carrier signals.
X0	Output	This outputs the contents of the X register.
RPUA~D	Input	Setting these pins to HIGH sets internal pull-up resistance for the corresponding R input/output port.

7. Absolute maximum constants (Ta=25℃)

Item	Symbol	Maximum rating	Unit
Power supply voltage	V_{DD}	−0.3~7.0	V
Power dissipation	Pd	700*1	mW
Operating temperature range	Topr	−10~+70	℃
Storage temperature range	Tstg	−55~+125	℃

*1 Reduced by 7mW for each increase in Ta of 1℃ over 25℃.

8. Electrical characteristics (Unless otherwise noted, Ta = 25℃, V_{DD1} = 3.0V, V_{DD2} = 5.0V)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Conditions
$\overline{INIT}$ input current (HIGH)	I_{IHIT}	—	0.1	5	μA	$V_I=V_{DD1}$
$\overline{INIT}$ input current (LOW)	I_{ILIT}	—	15	—	μA	V_I=GND
K input current (HIGH)	I_{IHK}	—	0.1	5	μA	$V_I=V_{DD1}$
K input current (LOW)	I_{ILK}	—	40	—	μA	V_I=GND
K input voltage (HIGH)	V_{IHK}	2.1	—	3	V	—
K input voltage (LOW)	V_{ILK}	0	—	0.9	V	—
R input voltage (HIGH)	V_{IHR}	2.1	—	3	V	—
R input voltage (LOW)	V_{ILR}	0	—	0.9	V	—
RHC input voltage (HIGH)	V_{IHRHC}	2.1	—	3.0	V	—
RHC input voltage (LOW)	V_{ILRHC}	0	—	0.9	V	—
RPU input voltage (HIGH)	V_{IHRPU}	2.1	—	3.0	V	—
RPU input voltage (LOW)	V_{ILRPU}	0	—	0.9	V	—
KHC input voltage (HIGH)	V_{IHKHC}	2.1	—	3.0	V	—
KHC input voltage (LOW)	V_{ILKHC}	0	—	0.9	V	—
R input current (HIGH)	I_{IHR}	—	0.1	5	μA	$V_I=V_{DD1}$
R input current (LOW)	I_{ILR}	—	40	—	μA	V_I=GND
R output voltage (LOW)	V_{OLR}	—	0.2	—	V	I_{OL}=1mA
R output leakage current	I_{LR}	—	—	5	μA	$V_O = V_{DD1}$, output Tr: OFF
XO output voltage (HIGH)	V_{OHXO}	—	2.5	—	V	I_{OL}=−1mA
XO output voltage (LOW)	V_{OLXO}	—	0.2	—	V	I_{OH}=1mA
R/$\overline{IR}$ input voltage (HIGH)	V_{IHRIR}	2.1	—	3.0	V	—
R/$\overline{IR}$ input voltage (LOW)	V_{ILRIR}	0	—	0.9	V	—
HALT input voltage (HIGH)	V_{IHHALT}	2.1	—	3.0	V	—
HALT input voltage (LOW)	V_{ILHALT}	0	—	0.9	V	—
I input voltage (HIGH)	V_{IHI}	3.5	—	5.0	V	—
I input voltage (LOW)	V_{ILI}	0	—	1.5	V	—
KHC input current (HIGH)	I_{IHKHC}	—	—	1	μA	$V_I=V_{DD1}$
KHC input current (LOW)	I_{ILKHC}	—	—	−1	μA	V_I=GND
R/$\overline{IR}$ input current (HIGH)	I_{IHRIR}	—	—	1	μA	$V_I=V_{DD1}$
R/$\overline{IR}$ input current (LOW)	I_{ILRIR}	—	—	−1	μA	V_I=GND
HALT input current (HIGH)	I_{IHHALT}	—	—	1	μA	$V_I=V_{DD1}$
HALT input current (LOW)	I_{ILHALT}	—	—	−1	μA	V_I=GND
I input current (LOW)	I_{ILI}	—	—	−1	μA	V_I=GND
PC, PA input voltage (HIGH)	V_{OHP}	—	2.5	—	V	I_{OH}=−1mA

Parameter	Symbol	Min.	Typ.	Max.	Unit	Conditions
PC, PA output voltage (LOW)	V_{OLP}	—	0.2	—	V	I_{OL}=1mA
A output voltage (HIGH)	V_{OHA}	—	2.5	—	V	I_{OH}=−1mA
A output voltage (LOW)	V_{OLA}	—	0.2	—	V	I_{OL}=1mA
Y output voltage (HIGH)	V_{OHY}	—	2.5	—	V	I_{OH}=−1mA
Y output voltage (LOW)	V_{OLY}	—	0.2	—	V	I_{OL}=1mA
ϕ output voltage (HIGH)	$V_{OH\phi}$	—	2.5	—	V	I_{OH}=−1mA
ϕ output voltage (LOW)	$V_{OL\phi}$	—	0.2	—	V	I_{OL}=1mA
WDT output voltage (HIGH)	V_{OHWDT}	—	2.5	—	V	I_{OH}=−1mA
WDT output voltage (LOW)	V_{OLWDT}	—	0.2	—	V	I_{OL}=1mA
S output voltage (LOW)	V_{OLS}	—	0.2	—	V	I_{OL}=1mA
OUT2 output voltage (HIGH)	V_{OHOUT2}	—	2.5	—	V	I_{OH}=−1mA
OUT2 output voltage (LOW)	V_{OLOUT2}	—	0.2	—	V	I_{OL}=1mA
SOC input voltage (HIGH)	V_{IHSOC}	$0.7V_{DD1}$	—	V_{DD1}	V	—
SOC input voltage (LOW)	V_{ILSOC}	0	—	0.9	V	—
OUT output voltage (HIGH)	V_{OHOUT}	—	2.5	—	V	I_{OH}=8mA
OSC2 output voltage (LOW)	V_{OLOSC}	—	0.5	—	V	I_{OL}=160 μA
OSC2 output voltage (HIGH)	V_{OHOSC}	—	2.5	—	V	I_{OH}=160 μA
S output leakage current	I_{LS}	—	—	5	μA	V_O = V_{DD1}, output Tr: OFF
OUT output current (HIGH)	I_{OHOUT}	—	25	—	μA	V_O = V_{DD1}, output Tr: OFF
OSC1 feedback current	I_{OSC1}	—	3	—	μA	V_{OSC1} = V_{DD1}
Static current consumption	I_{DDST}	—	—	10	μA	In standby state
Operating current consumption	I_{DDOP}	—	0.3	—	mA	fck=455kHz
Operating frequency	f_{OPR}	300	—	1000	kHz	—

2) EV2418B evaluation board for the BU2458 series

1. Overview

The EV2418B is an evaluation board for use with the BU2458 series of 4-bit, single-chip microcomputers. External attachment of a program ROM (2732, 2764, 27128, or 27256) on the BU2494 evaluation IC provides the same functions as the BU2458 series. This board can be used for system debugging of target systems.

2. Basic specifications

●A PROM is used as the program memory (2732, 2764, 27128, 27256, or equivalent).

●A power supply of 5V is supplied to the PROM, while power is supplied to the BU2494 from the target system.

●Input and output specifications are basically the same as those of the BU2458 series.

●The instruction system is the same as that of the BU2458 series.

●Operation notes

1 : Initial reset circuit

(1) Using the internal Power On Reset circuit

The BU2458 series is equipped with an internal Power On Reset circuit. The Power On Reset function works even if the $\overline{INIT}$ pin is in open state. For information on the characteristics of the reset signal timing, please refer to Figure 23.

(2) Using an external capacitor

If a longer reset timing is required than that which can be provided with the internal Power On Reset circuit, such as for power supplies with a slow rise time and power supplies with strong chattering ($>$1ms at fosc = 455kHz), a capacitor Co is connected between the $\overline{INIT}$ pin and the GND pin.

The CR integrated circuit is comprised of Co and the internal pull-up resistance of the IC (approximately 400 k). When the power is turned on, the integrated waveform of the power supply rise waveform is input through the $\overline{INIT}$ pin, and a reset is initiated which remains effective until the input threshold voltage of the internal inverter is exceeded. During the period in which the reset is valid, a clock with a stable cycle exceeding four machine cycles (24 clock pulses) must be input to the OSC1 pin.

The constant should be set high enough to accommodate the power supply rise time (including chattering) and the oscillation rise time subsequent to reaching the valid power supply voltage (the rise time for a crystal resonator is longer than that for a ceramic resonator).

If the power supply rise time is within 10ms, 0.1 μF is an appropriate value for Co.

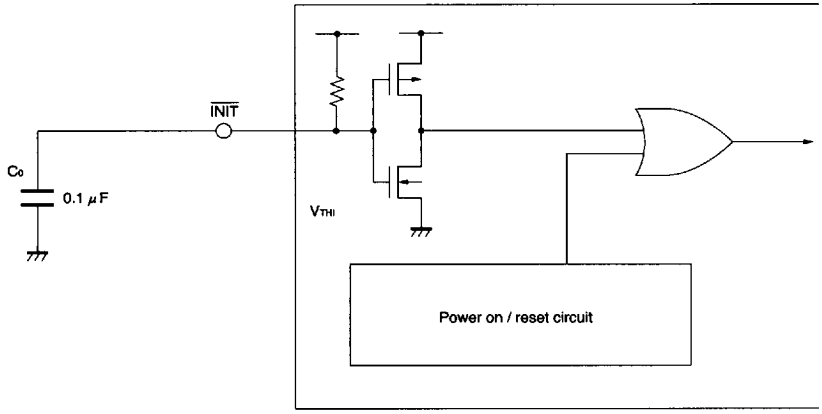


Fig. 23 Initial reset circuit

2. Oscillation circuit

Attaching an external ceramic (crystal) resonator between OSC1 and OSC2 enables configuration of a ceramic (crystal) oscillation circuit. In order to minimize effects on the board wiring and other elements, the resonator should be placed as close to OSC1 and OSC2 as possible.

The circuit shown in Figure 24 can be used by using a mask option (internal capacitor).

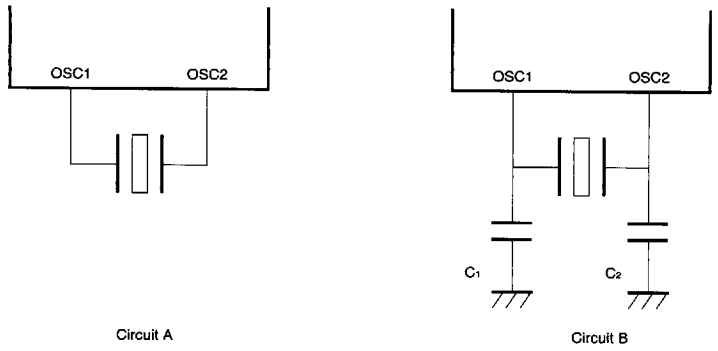


Fig. 24 Example of oscillation circuit (configuration)

a : Precautions regarding oscillation when the power is turned on

The setting should ensure that the valid $\overline{INIT}$ signal is longer than the power supply and oscillation rise times. If operation begins before the oscillation rise time has been completed, erroneous operation may occur. Generally, crystal resonators have a longer rise time than ceramic resonators, so special caution is required if using a crystal resonator.

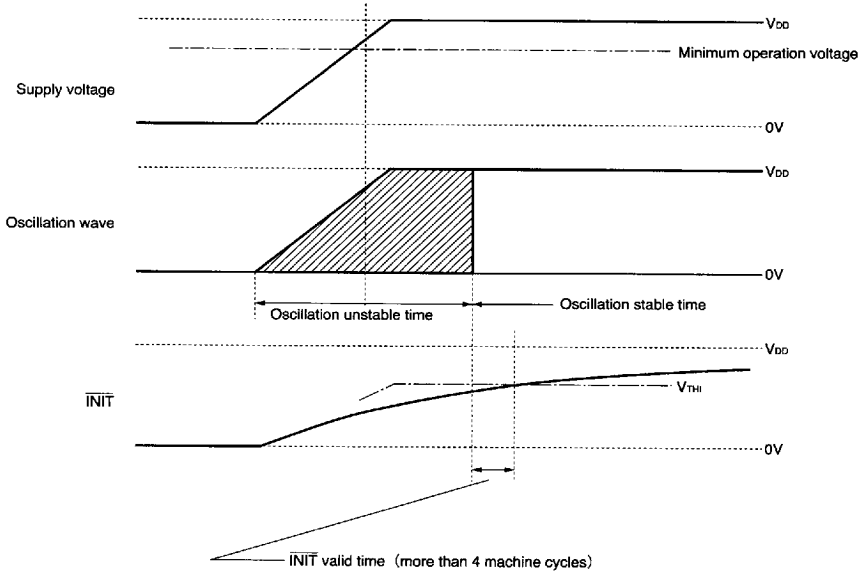


Fig. 25 Power supply waveform when power is turned on (expanded on time axis)

b : Precautions regarding oscillation when HALT is cancelled

When a HALT state is cancelled, oscillation is resumed. At this point, in order to assure sufficient time for the oscillation to stabilize, the BU2458 series is provided with an internal wait timer (6144 effective clock pulses / fosc) based on the hardware counter. During that period, no clock pulses are supplied to internal circuits, so no program execution is carried out. Consequently, input which is faster than the wait timer timing cannot be read.

The oscillation rise time varies depending on the type of resonator used. Please consult the manufacturer of the resonator for specific time information.

c : Correspondence table for resonators and application circuits

Currently, the resonators listed below can be used with the circuits and oscillation constants given in the table.

For more detailed information, please consult the manufacturer of the resonator before selecting a certain circuit and constant.

fosc=400kHz

Prod. No.	Maker	Circuit	Osc. constant	Power supply voltage range
CSB400P	Murata Mfg.	A	—	2.0~4.0
KBR400BK70	Kyocera	A	—	2.0~4.0
FCR400K3	TDK	A	—	2.0~4.0
EFO-A-400K04A	Matsushita Electric	A	—	2.0~4.0

fosc=440kHz

Prod. No.	Maker	Circuit	Osc. constant	Power supply voltage range
CSB440E	Murata Mfg.	A	—	2.0~4.0
KBR440BK70	Kyocera	A	—	2.0~4.0
EFO-A-440K04A	Matsushita Electric	A	—	2.0~4.0

fosc=455kHz

Prod. No.	Maker	Circuit	Osc. constant	Power supply voltage range
CSB455E	Murata Mfg.	A	—	2.0~4.0
KBR455BK70	Kyocera	A	—	2.0~4.0
FCR455K3	TDK	A	—	2.0~4.0
EFO-A-455K04A	Matsushita Electric	A	—	2.0~4.0

fosc=480kHz

Prod. No.	Maker	Circuit	Osc. constant	Power supply voltage range
CSB480E	Murata Mfg.	A	—	2.0~4.0
KBR480BK70	Kyocera	A	—	2.0~4.0
FCR480K3	TDK	A	—	2.0~4.0
EFO-A-480K04A	Matsushita Electric	A	—	2.0~4.0

fosc=432kHz

Prod. No.	Maker	Circuit	Osc. constant	Power supply voltage range
CSB432E	Murata Mfg.	A	—	2.0~4.0

fosc=320kHz

Prod. No.	Maker	Circuit	Osc. constant	Power supply voltage range
CSB320D	Murata Mfg.	B (Note)	C1=100pF C2=100pF	2.0~4.0

(Note) An internal capacitor and external capacitor are required.

4 : Precautions regarding setting of mask options
When setting the mask options for the BU2458 series, the following precautions should be observed.

a : Pull-up resistance
When pull-up resistance is used as input, always use the internal pull-up resistance (Y), or make sure that a HIGH or LOW state is input from an external source. An unstable level can cause erroneous operation because of noise and other problems, as well as increased current consumption.

b : HALT cancel function
When applying a HALT (oscillation stopped) state, at least one pin should be set to the HALT cancel function YES (Y) state, as a means of cancelling the HALT state. The HALT state will not be cancelled if a LOW state is input to a pin for which the HALT cancel function has been set to NO (N).
A HALT state can also be cancelled by inputting a valid reset signal.

c : S6 and S7 output states in HALT state
In applications where the HALT state is cancelled by shorting an input pin for which the HALT cancel function (see above) has been set (for example, with remote control transmitters), select (A), which produces a LOW state.
This ensures that the HALT state will be cancelled. (The pin will be LOW regardless of the program. However, the state is maintained internally, so recovery occurs following the cancel of the HALT state.)
If pull-up resistance has been added externally, current flows during a HALT state (output is LOW), so other means of retaining the values prior to the HALT state are required, such as selecting (B).

d : Resetting the watchdog timer using the OUT output signal
In applications where the OUT output signal is output periodically, such as in a remote control transmitter, selecting (Y), which resets the timer, can be used effectively.
If (N) is selected, so that the timer is not reset, the program must be designed so that the WDTR instruction is executed periodically (within 2 13 machine cycles).

e : Capacitors for ceramic resonator circuits
Select whether the capacitor is internal (Y) or not (N), depending on the resonator and oscillation circuit being used.

5 : Caution when developing programs
When developing programs for the BU2458 series, caution is required concerning the following items.

a : All of the instructions listed below must be stored in the ROM. (As long as they are in the ROM, they do not necessarily have to be used in the software.)

CAL	CLA	IYC	LAK	LMAIY
LYA	LYM	LAM	NOP	RTN
SED	REO	HALT	WDTR	LAY
LMA	IA	DAN	DYN	LAR
LXC	CMS			

(Note) Of the instructions listed above, those requiring operands must have appropriate values added as the operands.

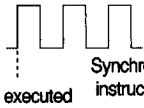
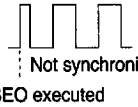
b : The following instructions should not be used as program start addresses (PAGE = 0, PC = 0).
BR SEO REO CAL RTN

c : When the OUT port is HIGH output
①If Y - REG = 8, the SEO instruction should not be executed.
②Do not execute the CMS instruction.

6 : Precautions when evaluating programs

Mass-produced chips can produce operations equivalent to those of evaluation chips (evaluation board : 2418B), but the table below lists differences between the two. If evaluation chips are being used, evaluation should be carried out keeping these items in mind.

Also, other fundamental characteristics are similar, but a sample should be used and testing carried out on mass-produced boards, to make sure the characteristics of final attachment circuits and other components match.

Class	Item	Mass-produced chips (BU2458 series)	Evaluation chip (EV2418 evaluation board)
Option settings	K input	HALT cancel function can be selected individually for each bit, using mask option.	HALT cancel function can be selected individually for each bit by setting HALT cancel function, using setting switches (KHC0 ~ KHC3).
	S output	Output state for S6 and S7 in HALT state can be selected, using mask option.	Output for S6 and S7 in HALT state can be selected using output state setting switches (SOC6, SOC7).
	R input/output	HALT cancel function can be selected individually for each bit, using mask option. Pull-up resistance can be selected individually for each bit, using mask option.	HALT cancel function can be selected individually for each bit by setting HALT cancel function, using setting switches (RHCA ~ RHCD). Pull-up resistance can be selected individually for each bit by setting pull-up resistance setting switches (RPUA ~ RPUD).
	Oscillation circuit	Oscillation capacitor can be added between GND pins, using mask option.	Supported by two types of evaluation chips (internal oscillation capacitor used/not used).
	WDT	Function to reset watchdog timer (WDT) using OUT output can be selected, using mask option.	WDT reset using OUT output can be selected, using WDT setting switches (R/IRB).
Electrical characteristics	Reset circuit	Internal Power On Reset circuit	No internal Power On Reset circuit. (External capacitor must be attached.)
	Power supply	Connected to 2.0 ~ 4.0 V power supply	[Power to EPROM supplied from separate power supply (5 V)] Connected to 2..5 ~ 4.0 V power supply. [Used with single power supply] Usable voltage range for EPROM must be taken into consideration. Be aware that current consumption increases.
	OSC input/output	Oscillation circuit can be configured by mounting ceramic resonator.	Mounting ceramic resonator enables configuration of oscillation circuit on evaluation board. (OSC pin on emulation cable side is open.)
		Note) The wiring capacitance and other characteristics are different from those of boards configuring the oscillation circuit, so the oscillation of mass-produced boards must be evaluated.	
Others	OUT output	Carrier signals are synchronized to rise of OUT output.  SEO executed Synchronized to SEO instruction	Short pulse may be output from carrier when OUT output rises.  Not synchronized to SEO instruction SEO executed

7 : Application circuit example

This shows a typical application example using the BU2458 as a base chip in a remote control encoder.

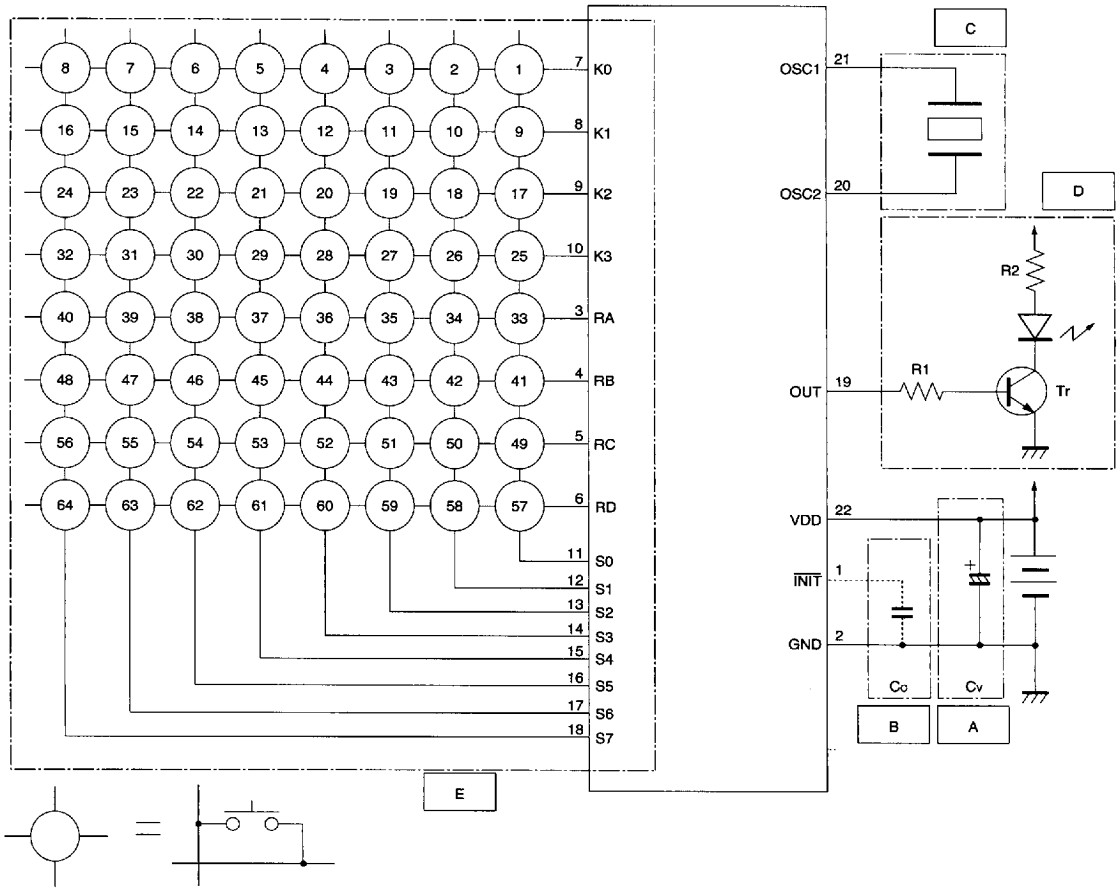


Fig. 26 Application circuit example (64-key remote control unit)

The following is a guide to peripheral circuits (A to E in the above diagram).

A: Power supply circuit	(page 37)
B: Initial reset circuit	(page 31)
C: Oscillation circuit	(pages 31 ~ 32)
D: Infrared LED drive circuit	(pages 37 ~ 38)
E: Key scan circuit	(pages 38)

(Note) There are no RC or RD pins in the BU2461 and BU2462.

●Operation notes

Although the application circuit example shwon above is recommendable, it is requested that the circuit characteristics should be confirmed carefully. When using the circuit with the external circuit constant changed, consider the marginal difference of the static characteristics and transient characteristics about the attachments including Rohm ICs.

Please note that we have not confirmed about the patent.

A : Power supply circuit

If large current flows to the circuit, such as in infrared LED drive circuits, and there is high wiring impedance on the board and high power supply impedance (in cases where the battery wears down particularly quickly because the equivalent impedance of the battery is too high), the power supply voltage fluctuates, and the IC may malfunction because it is operating outside of the operating power supply voltage range.

To suppress fluctuation in the power supply voltage and assure stable operation, an electrolytic capacitor should be connected between VDD and GND, as close to the IC pin as possible.

Depending on the settings entered for specifications such as the signal arrival distance and usage power supply voltage range, as well as conditions such as the type of battery used and the board design, a capacitance of 47 ~ 100 μF should be provided, with the appropriate value being determined through experimentation. If it is not possible to attach a large capacitor sufficiently close to the IC pin because of the mounting space or other considerations, a smaller capacitor (0.01 ~ 0.1 μF) may be added in close proximity to provide supplemental capacitance.

D : Infrared LED drive (OUT pin application) circuits

An infrared LED drive circuit can be configured by adding a single external NPN Tr to the OUT output.

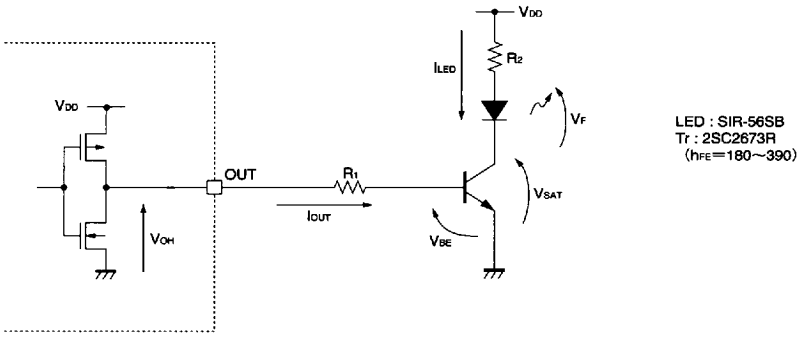


Fig. 27 Example of OUT pin attachment circuit

The following show examples of how the various constants in the circuit in Fig. 27 are determined.

a : Setting the LED drive current (I_{LED})

This is set with the type of LED and the arrival distance of the infrared beam in mind.

Using an SIR-56SB as the LED, at V_{DD} = 3V, designing a circuit with a target value of I_{LED} = 700mA involves the following calculation examples.

b : Setting R_1

The Tr must be put in overdrive and a base current value must be set that cannot fail to cause saturation, taking dispersion and temperature characteristics into consideration.

Assuming an overdrive factor of K, the base current I_{OUT} is determined as shown in equation (1).

$$I_{OUT} = \frac{K \times I_{LED}}{h_{FE}} \quad \cdots\cdots\cdots (1)$$

In equation (1), if h_{FE} = 270 typ. (see Tr specifications) and I_{LED} = 700mA, then setting K = 5 produces the following value for I_{OUT}.

$$I_{OUT} = \frac{5 \times 700mA}{270} \fallingdotseq 13mA$$

Caution is required, because if K is excessively large, the accumulation time for Tr will be longer, resulting in a longer ON time.
Assuming an OUT output HIGH voltage of V_{OH} and a Tr base – emitter voltage of V_{BE}, resistance R_1 is determined by equation (2).

$$R_1 = \frac{V_{OH} - V_{BE}}{I_{OUT}} \quad \cdots\cdots\cdots (2)$$

In equation (2), if V_{OH} = 2.2V (at I_{OUT} = 13mA : see Figure 9) and the V_{BE} of Tr = 0.8V, the following equation results.

$$R_1 = \frac{2.2 - 0.8}{13 \times 10^{-3}} = 107.7\ (\Omega)$$

As a result, a value of $R_1 \fallingdotseq 110\ (\Omega)$ is appropriate. (At this point, it is assumed that I_{OUT} will fluctuate between 12.5mA and 14.0mA because of fluctuation of the IC.)

c : Setting R_2

Resistance R_2 is expressed by equation (3), assuming the saturation voltage when Tr is ON (I_{LED} = 700mA) is V_{SAT} and the forward voltage of the infrared LED is V_F.

$$R_2 = \frac{V_{DD} - V_F - V_{SAT}}{I_{LED}} \quad \cdots\cdots (3)$$

In equation (3), if V_{SAT} = 0.25V and V_F = 1.65V (see the specifications for Tr and LED), the following value results.

$$R_2 = \frac{3.0 - 1.65 - 0.25}{700 \times 10^{-3}} = 1.57\ (\Omega)$$

Based on this result, $R_2 \fallingdotseq 1.6\,\Omega$ is an appropriate value. (At this point, V_{SAT} fluctuates because of the base current value (I_{OUT}) of (1), but since there is sufficient saturation, the fluctuation of V_{SAT} can be suppressed to around 0.1V or less. (See the specifications for Tr.)

E : Key scan circuit

The key scan circuit is configured as shown in Figure 28. (S output - K input)

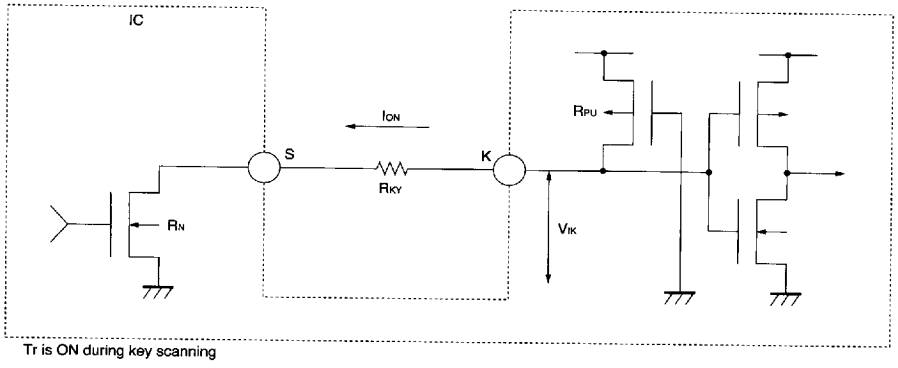


Fig. 28 Key scan circuit (equivalent circuit when key switch is ON)

When configuring a key scan circuit, the following precautions should be observed.

If a material with high ON resistance (R_{KY}) such as conductive rubber is used as the key switch, the input voltage V_{IK} to the K (R) pin will only drop as far as differential voltage between $R_{KY} + R_N$ and R_{PU} (see equation 4), even when a key is pressed. Key scanning cannot be carried out properly if this becomes higher than the K (R) input LOW voltage (V_{ILK}). Therefore, the design should ensure that R_{KY} stays lower than the value indicated below.

(R_{KY} indicates the resistance value between the two IC pins when the key is on, including the wiring resistance.)

The following calculation is carried out under the conditions V_{DD} = 3V and Ta = 25℃.

Assuming that $R_N \ll R_{KY}$ and R_{PU} :

$$V_{IK} = \frac{R_{KY}+R_N}{R_{PU}+R_{KY}+R_N} \times V_{DD} \fallingdotseq \frac{R_{KY} \times V_{DD}}{R_{PU}+R_{KY}} \times V_{DD} \quad \cdots (4)$$

$V_{IK} \leqq V_{IL}$ (max.) is the condition required in order for key scanning to be carried out correctly (the input pin voltage must be below the maximum input LOW voltage noted in the specifications). This gives us the following :

$$V_{IK} = \frac{R_{KY}}{R_{PU} + R_{KY}} \times V_{DD} \leqq V_{IL}\ (\text{Max.}) \quad \cdots (5)$$

In equation (5), a worst-case scenario is assumed in which R_{PU} will be the minimum value (see specifications), so the setting range which determines R_{KY} is as follows :

$$R_{KY} \leqq \frac{R_{PU}\ (\text{Min.}) \times V_{IL}}{V_{DD} - V_{IL}} \fallingdotseq \frac{60\text{k} \times 0.9}{3 - 0.9} = \underline{25.7\ (\Omega)}$$

In addition to the above, if keys are arranged in serial order and Di is inserted, please be aware that conditions will deteriorate further. We recommend that operation be evaluated using an actual board.

Also, aside from an increase in the input voltage caused by the resistance of the keys and wiring, if the circuit has a capacitance such as wiring, the effects of that capacitance must be taken into consideration as well.

This capacitance causes the rise and fall times of the key scan signals to be slower, throwing the key scan timing off far enough that the level does not reach the LOW state, or a HIGH state is not reached before the next input timing begins. These situations can make it impossible to obtain accurate key scan results. To prevent such problems, the program design must ensure that key pulses are output for a longer period of time than the delay time caused by the wiring impedance.

Also, if using a program developed by Rohm, the initial delay timing (a and b in Figure 29) of the finished program must be within 200 μs. (Rohm key scan programs are standardized to that timing.)

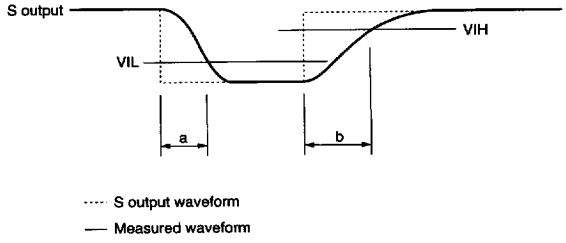


Fig. 29 Key scan signal and input timing

●Electrical characteristic curves
(Data indicate typical values and not guaranteed (rated) values.)

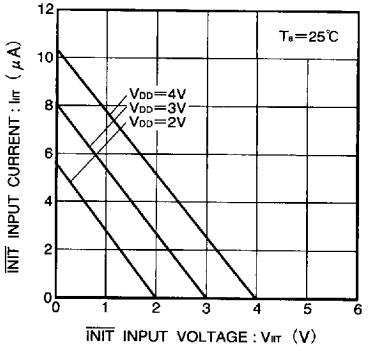


Fig. 30 $\overline{INIT}$ input current vs. voltage characteristic (typical values)

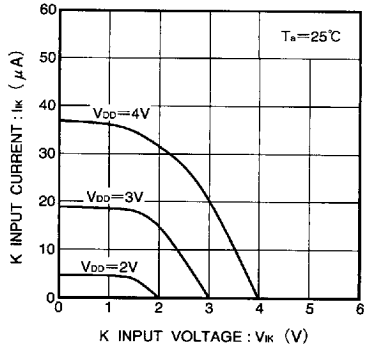


Fig. 31 K input current vs. voltage characteristic (typical values)

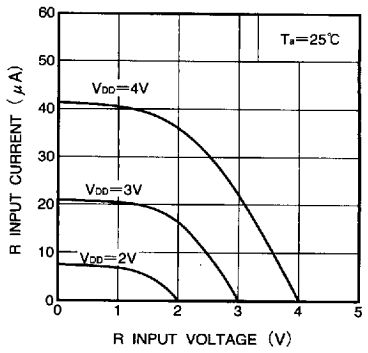


Fig. 32 R input current vs. voltage characteristic (typical values)

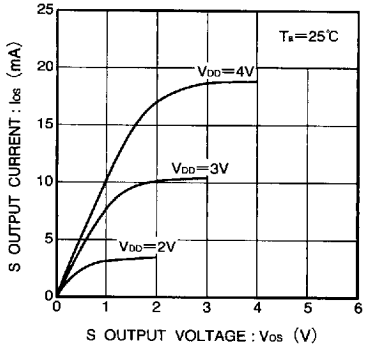


Fig. 33 S output current vs. voltage characteristic (typical values)

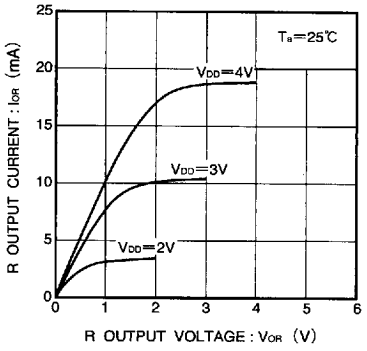


Fig. 34 R output current vs. voltage characteristic (typical values)

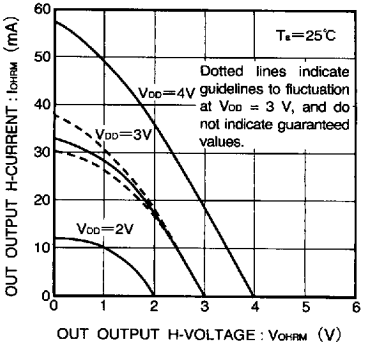


Fig. 35 OUT output HIGH current vs. voltage characteristic (typical values)

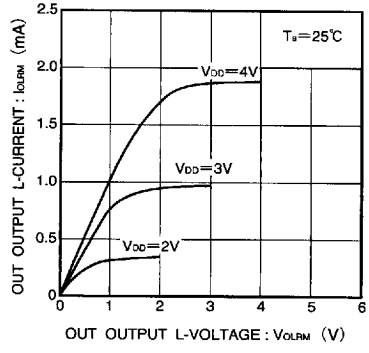


Fig. 36 OUT output HIGH current vs. voltage characteristic (typical values)

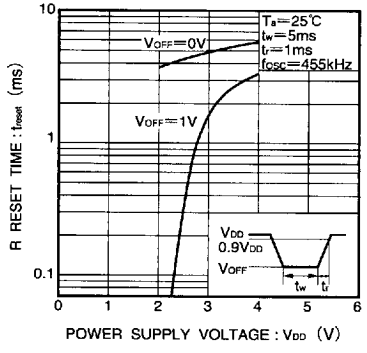
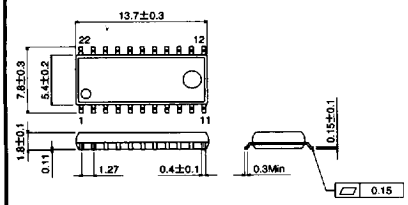


Fig. 37 Reset time vs. power supply voltage characteristic (typical values)

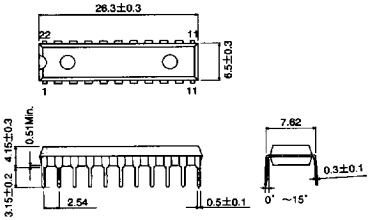
●External dimensions (Units: mm)

BU2458



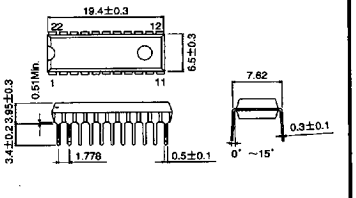
SSO22

BU2459



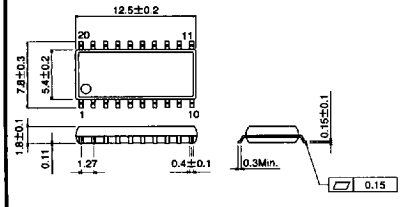
DIP22

BU2460



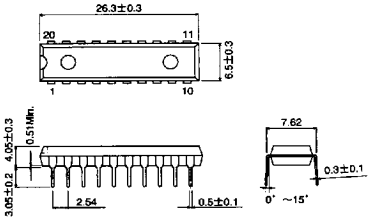
SDIP22

BU2461



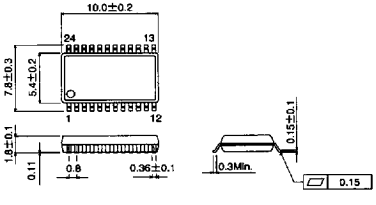
SOP20

BU2462



DIP20

BU2456



SSOP-A24