九齊科技股份有限公司
Nyquest Technology Co., Ltd.

DATA SHEET

NY9UP01A

Single Remote Controller with13 I/O

Version 1.4

Aug. 27, 2018

Revision History

Version	Date	Description	Modified Page
1.0	2016/11/23	Formal release.	-
1.1	2017/02/14	1. Adjust LVR to 1.6V. 2. Adjust detecting LVR occurrence times sample. 3. Adjust Timer IR generation description.	5, 12 18 20
1.2	2017/05/25	1. Modify the descriptions of Virtual LVD in Features. 2. Modify IR Instructions. 3. Modify Other Instructions.	5 34 35
1.3	2017/11/24	1. Adjust POR to 1.6V. 2. Adjust LVR to 1.4V. 3. Adjust Virtual LVD to 1.4V. 4. Add duty attention. 5. Add TX wave description.	5 5, 12 5 21 22, 23
1.4	2018/08/27	1. Adjust POR to 1.3V. 2. Adjust LVR to 1.3V. 3. Delete Virtual LVD description.	5 5, 12 17

Table of Contents

Chapter 1. Introduction

1.1 General Description

The NY9UP01A is a powerful 4-bit MCU with remote controller. It has 13 I/O ports and supports T-type key matrix. One large sink current IR port can transmit without any bipolar transistor. The RISC MCU architecture is very easy to use, and various applications can be easily implemented. There are 38 instructions, and most of them are executed in single cycle. Furthermore, it provides the HALT mode (sleep mode) to extend battery life.

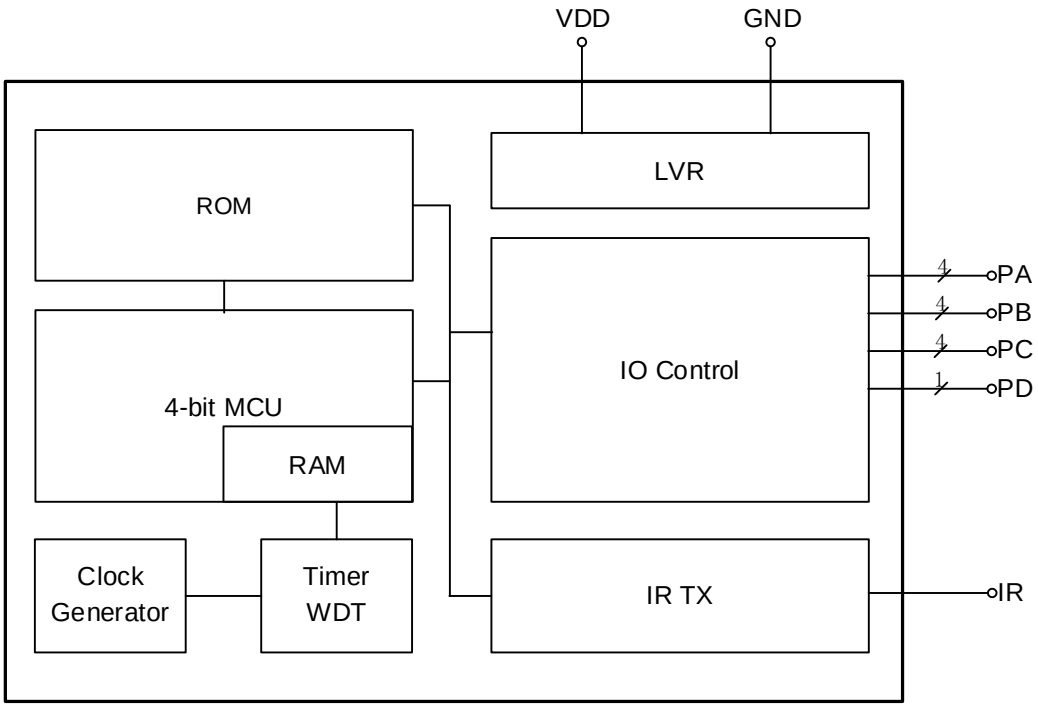
1.2 Features

- Operating voltage range: 2.0V to 3.6V.
- 4-bit RISC type micro-controller with 38 instructions.
- 768 x 10-bit program ROM.
- 32 x 4-bit RAM, indirect RAM addressing mode is supported.
- 1MHz instruction frequency.
- HALT mode to save power, standby current <1uA @3V.
- Precisely embedded oscillator with build-in resistor, +/- 1.0% deviation in 2.0V~3.6V and -20°C~+70°C.
- Power on reset (POR=1.3V).
- Low voltage reset (LVR=1.3V) and watch-dog reset both are supported to protect the system.
- One entrance for interrupt operation with an independent stack, multiple interrupt sources.
- 13 flexible I/Os of PAx, PBx, PCx and PD0 with optional function: initial output high, initial output low, bi-direction I/O with pull-high, bi-direction I/O without pull-high, initial pull-high input or initial floating input. *(Code option)*
- M-Type, T-type or mixed type key wakeup supported.
- IR provides TX application, optioned for 100%, 50% large current IR carrier output, or Normal (Sink/Drive) for TX.
- 8-bit readable timer with selectable timer clock source for IR TX carrier frequency.
- Security bit for code protection. (OTP cannot be read when Security bit is set)

1.3 Product List

IC Type	ROM (bits)	RAM (bits)	I/O	T-scan	LVR	8 bit Timer	IR TX
NY9UP01A	768 x 10 (OTP)	32 x 4	13	v	v	v	1

1.4 Block Diagram



1.5 Pad Description

Pad	ATT	Description
VDD	Power	Positive power
GND	Power	Negative power
IR	O	Infrared port (TX)
PA0/SDA	I/O	Bit 0 for Port A, or serial data input at programming mode.
PA1/SCL	I/O	Bit 1 for Port A, or serial clock input at programming mode.
PA2/Mode	I/O	Bit 2 for Port A, or select programming mode.
PA3/Vpp	I/O	Bit 3 for Port A, or positive high power for programming.
PB0~3	I/O	PA0~3, PB0~3, PC0~3, PD0: 13 flexible I/Os with optional function. Port can be set as normal I/O or key scan I/O, and key scan I/O can send key scan signal under halt mode.
PC0~3	I/O	
PD0	I/O	

1.6 Electrical Characteristics

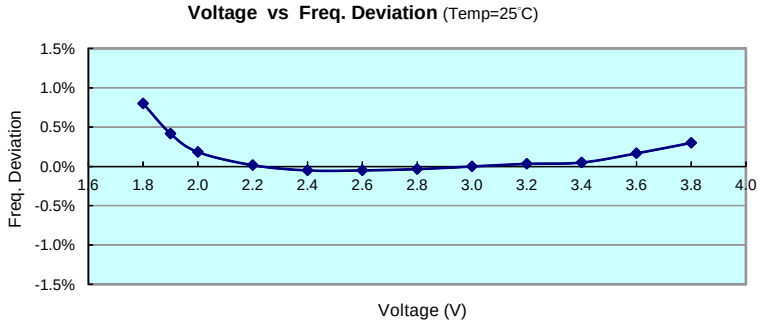
1.6.1 Absolute Maximum Rating

Symbol	Parameter	Rated Value	Unit
V_{DD} - V_{SS}	Supply voltage	-0.3 ~ +4.0	V
V_{IN}	Input voltage	V_{SS}-0.3V ~ V_{DD}+0.3	V
T_{OP}	Operating Temperature	-20 ~ +70	°C
T_{ST}	Storage Temperature	-40 ~ +85	°C

1.6.2 DC Characteristics *(V_{DD}=3.0V, T_A=25ºC, unless otherwise specified)*

Symbol	Parameter		Min.	Typ.	Max.	Unit	Condition
V_{DD}	Operating voltage		2.0	3	3.6	V	1 MHz
I_{SB}	Supply current	Halt mode			1	uA	Sleep, no load
I_{Scan}		Scan mode			2	uA	T-type key scan
I_{OP}		Operating mode		1		mA	1MHz, no load
V_{IH}	Input high level			0.7*V_{DD}		V	
V_{IL}	Input low level			0.5*V_{DD}		V	
I_{IL}	Input current (Internal 125KΩ pull-high)			24		uA	V_{IL} = 0V
I_{OH}	Output high current			-9		mA	V_{OH} = 2.0V
I_{OL}	Output low current			18		mA	V_{OL} = 1.0V
I_{IR}	IR sink current			450		mA	V_{IR} = 1.5V
△F/F	Frequency lot deviation		-1.0		1.0	%	V_{DD}: 2.0V ~ 3.6V, Temp: -20°C ~ +70°C

1.6.3 Voltage vs. Frequency



Chapter 2. Hardware Architecture

2.1 Overview

2.1.1 Function Block Diagram

The NY9UP01A belongs to the SRC family of the NYQUEST devices. Block diagram of the device is shown below.

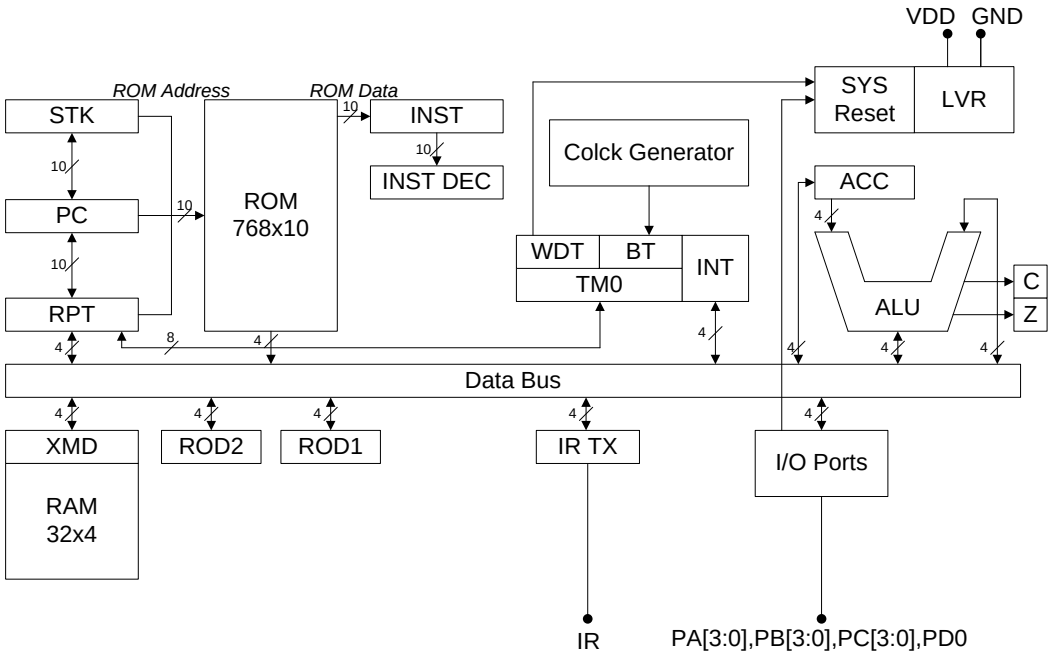


Figure 2-1 NY9UP01A Function Block Diagram

2.1.2 Arithmetic Logic Unit

The NY9UP01A provides a 4-bit arithmetic logic unit with a 4-bit accumulator to perform logic, unsigned arithmetic, data transfer and conditional branch operation. There are two flags (carry and zero) to indicate the result of the operation. One or two operands will be the data sources of the ALU operation. The operands can be ACC, RAM, register, or literal constant data.

2.1.3 ALU Related Status Flag

Besides RSTC and SETC commands directly assign the value of the carry flag, C is influenced by the arithmetic result. C means carry and also means the complement of borrow. If the addition operation larger than 0xF, C=1, and C=0 if the result ≤15. If the subtraction operation smaller than 0, C=0, and C=1 if the result ≥0.

Symbol	Flag	Description
C	Carry flag	C=1 if a carry-out occurs after an addition operation.
		C=0 if a borrow-in occurs after a subtraction operation.
Z	Zero flag	Z=1 if the result of an ALU operation is zero.

2.1.4 Address Pointer

The NY9UP01A micro-controller contains a program counter (PC), an interrupt dedicated stack (STK), and a multi-function register pointer (RPT). The length of each address pointer is 10-bit. Users have to keep in mind that the initial value of all the pointers is unknown, except the PC.

2.1.5 Program Counter (PC)

As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC starts from the reset vector (address 0x0000) after the system reset, and its value is increased by one every instruction cycle unless changed by an interrupt or a branch instruction. The interrupt vector is at address 0x0010.

2.1.6 Stack (STK)

One level hardware push/pop stack dedicated to the interrupt is available. When an interrupt occurs, the system pushes the PC to the STK automatically. When the program returns to the main program from the interrupt routine by IRET instruction, the system pops the STK back to the PC.

2.1.7 Multi-function Register Pointer (RPT)

As implied in the name, RPT are multi-function registers. Users have to operate RPT in coordination with instructions below.

Inst./Event	Function
CALL	Pushes PC+2 to RPT.
LDPC	Loads RPT to PC.
RBPC	Reads back PC+1 to RPT.
LDTM	Loads RPT to TM.
RBTM	Reads back TM counter to RPT.
RBRO	Use RPT as address to read ROM data.
XMD	Use RPT[8:0] as address to access indexed RAM data.

2.2 Memory Organization

The NY9UP01A has 768 words ROM, 32 nibbles of RAM and 16 nibbles of dedicated system control register. The registers are divided into 10 nibbles of system registers and 6 nibbles of memory registers. Besides, there are several registers without address allocation, and they can only be accessed by the special instructions such as clock source register (TCS) and timer register (TM).

2.2.1 ROM

A program data single ROM is provided and its structure is shown below. The reserved region contains system information and can't be utilized by users.

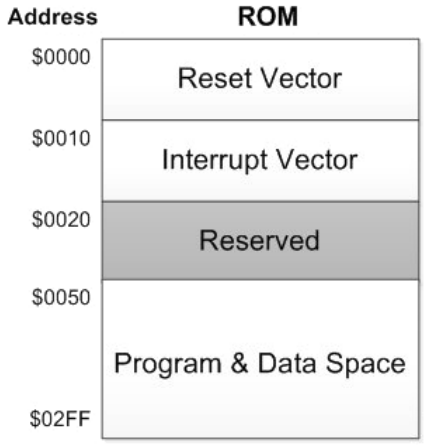


Figure 2-2 NY9UP01A ROM Map

2.2.2 RAM

NY9UP01A provide only 1 page, and the memory space is shared with the memory registers (address=0x00~0x07), the address for RAM is 0x20~0x3F.

Figure 2-3 NY9UP01A RAM Map

In addition to the immediate addressing mode, the indexed addressing mode is also supported. The page and address of the indexed RAM should be stored into RPT1 and RPT0 first, and users can read from or write in the XMD memory register to realize the indexed RAM access.

2.2.3 Memory Register

The 6 memory registers share the address with RAM. The page number of RAM has no relationship with the memory register address.

Address	Name	Bit	Description
0	RPT0	[3:0]	Multi-function register pointer bit [3:0]
1	RPT1	[3:0]	Multi-function register pointer bit [7:4]
2	RPT2	[1:0]	Multi-function register pointer bit [9:8]
3	--	--	Reserved
4	--	--	Reserved
5	XMD	[3:0]	Indexed RAM data access register
6	ROD1	[3:0]	ROM data bit [7:4] access register
7	ROD2	[1:0]	ROM data bit [9:8] access register

2.2.4 System Register

The NY9UP01A provides only two instructions to access the system registers. MVTA reads the register value of the address operand `t' to the ACC. MVAT writes the ACC value to the register of the address `t'.

Address	Name	Bit	Description
0x0	--	--	Reserved
0x1	DTLN	[3:0]	Duty Low nibble
0x2	DTHN	[3:0]	Duty High nibble
0x3	INT	[3:0]	Interrupt control register
0x4	TMC	[2:0]	Timer and interrupt flag control register
0x5	LVD	[1]	LVR/POR flag register
0x6	IRTX	[1:0]	TX control register
0x7	--	--	Reserved
0x8	PA	[3:0]	PA I/O port control register
0x9	PB	[3:0]	PB I/O port control register
0xA	PC	[3:0]	PC I/O port control register
0xB	PD	[0]	PD I/O port control register
0x#	Others	[3:0]	Reserved

2.2.5 Register without Memory Allocation

Name	Description	Instruction
TCS	2-bit timer clock source register of IR carrier	CHTCS
TM	8-bit IR carrier timer	LDTM

2.3 System Reset

2.3.1 System Power-On & Power-Down

After Power-on, the power-on reset initialization will automatically be set out. The system takes about 16ms to leave from the reset initialization procedure, and enters the normal operation and the program counter (PC) will start at the reset vector to execute the desired program.

2.3.2 Low Voltage Reset & Detection (LVR)

When the system enters the normal operation, the power voltage must be kept in an effective working voltage range. If the power voltage is lower than the effective working voltage range, the system will work improperly.

To prevent the system crash, NY9UP01A series supplies Low Voltage Reset (LVR) detectors. Once the LVR detector detects a harmful low voltage supply, it will cause a low voltage reset. The so-called “low voltage reset” point of the NY9UP01A IC is about 1.3V.

2.3.3 Watch-Dog Timer (WDT)

To recover from program malfunction, the NY9UP01A IC supports an embedded watch-dog timer reset. The WDT function always works with the program executing. Users have to clear the WDT periodically to prevent from timing up with a reset generation. Typically, the minimum time-up period of the WDT is about 0.45s. Users can move a 0xE value to the INT system register to clear WDT.

2.4 I/O Ports

There are 13 I/O ports, designated as PAx, PBx, PCx, PD0, and x=0~3. These ports can be configured by option code and I/O register, but the option code is fixed, and users can set I/O functions by I/O registers. The I/O ports can be configured as six statuses, initial output high, initial output low, I/O (bi-direction) with pull-high, I/O (bi-direction) without pull-high, initial pull-high input or initial floating input by option. And when the chip is running, the status can be changed by I/O register control.

The table below shows the relation between them. (PX means PA, PB, PC register)

Category	Option	PX register write 1	PX register write 0
PXx (X=A~C, x=0~3)	initial output high	output high	output low
	initial output low	output high	output low
	I/O with pull-high	input with pull-high	output low
	I/O without pull-high (open drain)	Input floating	output low
	initial pull-high input	input with pull-high	input floating
	initial floating input	input with pull-high	input floating

The pull-high resistor of all the I/O ports is about 125KΩ @3V for key matrix function usually.

2.5 Infrared Transmitter

The NY9UP01A provides an independent pin (IR) for infrared transmit block, which is used to send infrared signal. For the function of transmitter, users can set a variety of IR carrier frequency by the given clock source (TCS), DTHN, DTLN and the given value of 8-bit IR timer register (TM). As for the detailed calculation of IR carrier frequency and applications, refer to section 3.8, IR control register.

2.6 Interrupt Generator

There is one hardware interrupt and it has 2 different sources in NY9UP01A. The interrupt event can be a fixed interval of the system base timer (BT), or the timer overflow flag (TOF). The TOF can be selected as one of the sample rate timer overflow by the register INT. There is a system base timer in the NY9UP01A IC. The NY9UP01A provide 4 fixed intervals from the system base timer for interrupt source: 0.128ms, 0.256ms, 0.512ms and 1.024ms.

As an interrupt occurs, NY9UP01A stores the accumulator (ACC), carry flag (C), zero flag (Z) automatically. Then move PC to STK, and jump to the interrupt vector (0x0010). An interrupt routine finishes with an IRET instruction. The IC draws the ACC, C, Z and PG back, and moves STK to PC back to jump back the main program.

The interrupt event of BT will be automatically cleared after entering the interrupt routine, but the TOF have to be cleared by users.

Chapter 3. System Control Register

3.1 Introduction of System Control Register

The TCS, TMC and TM are IR carrier control related registers. The PA PB PC and PD are I/O ports registers. INT register is used to control or access the BT, TOF and WDT. The combination of RPT0~2 are multi-function register pointer. The C and Z are arithmetic associated flags. The XMD are RAM access registers. The ROD1 and ROD2 registers are used to read the ROM data.

3.1.1 System Control Register Address Map

Addr	Name	RSTV	R/W	Bit	Data	Description
0	-	-	-	-	-	Reserved
1	DTLN	0000	R/W	[3:0]		Duty Low nibble, set carrier duty low 4 bits value at TX mode
2	DTHN	0000	R/W	[3:0]		Duty high nibble, set carrier duty high 4 bits value at TX mode
3	INT	xxxx	R	[0]	0/1	System base timer 0.128ms
				[1]	0/1	System base timer 0.256ms
				[2]	0/1	System base timer 0.512ms
				[3]	0/1	System base timer 1.024ms
			W	[3:0]	0000	Interrupt ~= 0.128 ms
					0001	Interrupt ~= 0.256 ms
					0010	Interrupt ~= 0.512 ms
					0011	Interrupt ~= 1.024 ms
					0100	Interrupt = TOF
					0101	Reserved
					0110	Interrupt OFF
					0111	Interrupt ON
					1000	Reserved
					1110	Clear WDT
					Others	Reserved
4	TMC	0000	R/W	[0]	0/1	TOF: Write0(clear)only; TM ON to activate TO Flag
				[1]	0/1	Reserved
				[2]	0/1	Reserved
				[3]	0/1	TMEN: TM will keep last value if ON after OFF without LDTM

Addr	Name	RSTV	R/W	Bit	Data	Description
5	LVD	0011	-	-	-	Reserved
			R/W	[1]	0/1	PORF: Write 0 (clear) only
			-	-	-	Reserved
6	IRTX	0001	R/W	[0]	0/1	TXLD: TX latch data, Write 1 to OFF TX
				[1]	0/1	LWC: Data L WO/W carrier,1: with carrier 0: without carrier
			-	[3:2]	-	Reserved
7	-	-	-	-	-	Reserved
8	PA	xxxx	R	[3:0]		Read port A input pad data
		0000	W	[3:0]		Write to port A data register
9	PB	xxxx	R	[3:0]		Read port B input pad data
		0000	W	[3:0]		Write to port B data register
A	PC	xxxx	R	[3:0]		Read port C input pad data
		0000	W	[3:0]		Write to port C data register
B	PD	xxxx	R	[0]		Read port D input pad data
		xxx0	W	[0]		Write to port D data register
#	Others	-	-	[3:0]		Reserved

3.1.2 Memory Register Address Map

Addr	Name	R/W	Bit	Description	Initial	Wake-up
0	RPT0	R/W	[3:0]	Multi-function register pointer [3:0]	0	U
1	RPT1	R/W	[3:0]	Multi-function register pointer [7:4]	0	U
2	RPT2	R/W	[3:0]	Multi-function register pointer [9:8]	X	U
3	-----	----	----	Reserved	-	-
4	-----	----	----	Reserved	-	-
5	XMD	R/W	[3:0]	Indexed RAM data access register	X	X
6	ROD1	R/W	[3:0]	ROM[7:4] data access register	X	U
7	ROD2	R/W	[1:0]	ROM[9:8] data access register	X	U

3.1.3 Register without Memory Allocation Map

Name	R/W	Bit	Description	Initial	Wake-up
C	-	1	Arithmetic carry flag	0	U
Z	-	1	Arithmetic zero flag	0	U
TCS	W	2	Timer clock source selection of TM	8MHz	U
TM	W	8	IR carrier timer	X	U

R : Can be read from the register

U : Unchanged (the same as before wake-up)

W : Can be written to the register

X : Unknown

3.2 RPT

The RPT of NY9UP01A is 10-bit, and the functions of RPT are listed in the section 2.2.3. Besides the instructions related to the XMD only access bit [5:0] of the RPT, others access all available bits. The RPT will be frequently accessed because of its multi-functionality, so the NY9UP01A series provides 3 instructions to accelerate the access of RPT0~2: MVRM, MVMR and MVLR.

The CALL instruction pushes the PC to the RPT and jump to the subroutine address of the operand `a'. When the subroutine is finished, use LDPC to come back to the main program.

3.3 ROD

The NY9UP01A provides the RBRO instruction to read the ROM data out. When RBRO is executed, the system takes the RPT as ROM address, and the ROM data is loaded to ROD2, ROD1, and ACC. Bit [9:8] of the ROM data is loaded to ROD2, bit [7:4] to ROD1, and bit [3:0] to ACC. Using RBRO to read the data of the reserved ROM area out is unacceptable, and results in a reset. The RBRO instruction has a 1-bit operand `n'. 0 means the RPT value keeps unchanged after RBRO, and 1 means the RPT adds 1.

3.4 RAM Control Register

3.4.1 XMD

Users access XMD taking the RPT[5:0] as the address. Users have to watch out that the NY9UP01A does not support using XMD to access memory registers, so the RPT[5:0] can't be 0x0~0x7 when accessing XMD.

3.5 I/O Ports Register

Each I/O port has its corresponding register PX. Reading from the register reveals the pad status, and writing to it means writing to the I/O register.

The four registers of PA, PB, PC and PD can configure the corresponding port status, and it must cooperate with the port option. The detail can check the section 2.4.

The register PX of an input port is used for pull-high on/off. The register of a bi-direction port is used for switch the I/O between input and output. When the register PX=1, it is an input with or without pull-high. When PX=0, it is an output and output low level. The register PX value of an output port simply means the output data. Users have to note that reading from an output port also getting the pad potential level, not the register value.

The pull-high resistor of all the I/O ports is only the strong pull-high, which is about 125KΩ @3V for key matrix function usually.

Example 3-1 Setting PA, PB, PC

```
; if Port A set option as I/O with pull-high
    MVLA   0x0
    MVAT   PA                   ; Set PA as output low
    MVLA   0xF
    MVAT   PA                   ; Set PA as input pull-high

; if Port B set option as I/O without pull-high(open drain)
    MVLA   0x0
    MVAT   PB                   ; Set PB as output low
    MVLA   0xF
    MVAT   PB                   ; Set PB as input floating

; if Port C set option as initial output high
    MVLA   0x0
    MVAT   PC                   ; Set PC as output low
    MVLA   0xF
    MVAT   PC                   ; Set PC as output high

; if Port C set option as initial input pull high
    MVLA   0x0
    MVAT   PC                   ; Set PC as input floating
    MVLA   0xF
    MVAT   PC                   ; Set PC as input pull high
```

3.6 LVD Register

The LVD register is used to recognize the status of supply voltage. The LVD[1] is POR Flag and can't set to 1 by any reset sources but POR. Even the voltage is lower than Vlvr to cause LVR, the POR flag won't be forced to high unless the power disappears. The flag can be cleared by setting 0 to its register.

Category	Bit	Description
LVD	[1]	PORF: Write 0 (clear) only

3.7 INT

The reading source and the writing destination of the system register of address 0x3 are different.

3.7.1 System Base Timer Polling

Reading the 4-bit data of INT acquires the value of the BT counter. The NY9UP01A series provides 4 different base timer intervals for polling: 0.128ms, 0.256ms, 0.512ms and 1.024ms, shown as Figure 3-1. The value of time means the period, so polling a data toggle means half time of the interval.

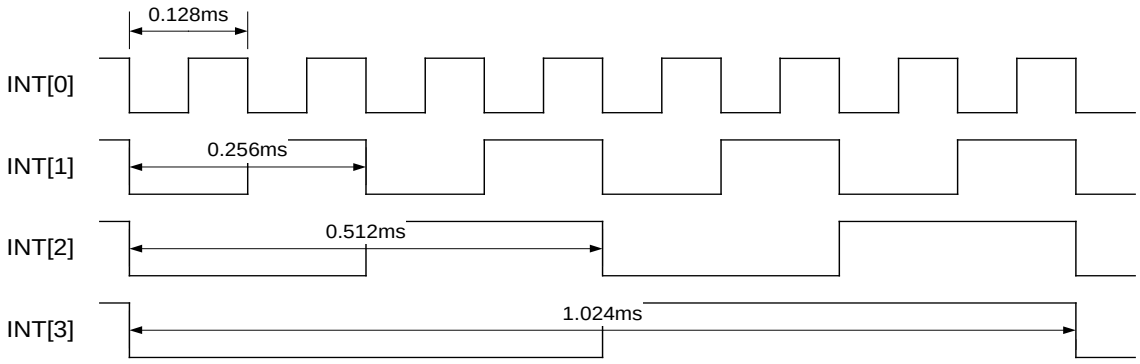


Figure 3-1 INT timing diagram

3.7.2 Interrupt Source

As mentioned in the section 2.6, the only one interrupt has 5 interrupt sources including 4 different BT intervals, and the TOF. Writing 0x0, 0x1, 0x2, 0x3, 0x4 to INT selects the interrupt source. Writing 0x6 to INT turns off the interrupt generator, and writing 0x7 to INT turns it on. Remember to set the source of interrupt before turning the interrupt on. If users want to change the interrupt source, turn off the interrupt first, set the source, and then turn it on.

3.7.3 Flag Clear

Writing 0xE to the INT clears the WDT.

Writing 0x0 to the TMC[0] clears the TOF.

3.8 IR Control Register

3.8.1 TCS

The TCS register indicates the TM clock source of IR carrier frequency. The different clock sources combined with a variety of value of the 8-bit timer registers will cause different IR carrier frequency. CHTCS loads the ACC to the TCS of the IR carrier, and the TCS can’t be read.

TCS	System	Description
0x0	1M	---
0x1	2M	---
0x2	4M	---
0x3	8M	Default

3.8.2 TM

The TM register includes a set of the 8-bit timer reload value latch and a set of the 8-bit downward counter. Users can load the latch and counter from RPT0 RPT1 by LDTM, then the counter counts down until to 0, next the counter is automatically reloaded from the latch again. With the different TM value, the IR carrier frequency will be affected correspondingly, and DTHN DTLN decides the duty of IR carrier. In addition, the timer will be stopped counting if the TM is loaded as 0 or timer is turned off (TMC[3] equals to 0).

LDTM loads the RPT to the TM of IR carrier frequency, and RBTM reads TM counter value to RPT. The table below shows the relationship between RPT, Timer, and Duty.

Timer A	7	6	5	4	3	2	1	0
RPT	RPT1[3:0]				RPT0[3:0]			
Duty	DTHN[3:0]				DTLN[3:0]			

In order to generate IR carrier, user need to set the 8-bit value of timer and duty at first. Duty bit[7:0] can be configured through register DTHN[3:0] & DTLN[3:0], then set TXLD = 0 and LWC = 1, and load RPT to Timer latch through LDTM instruction. At last need to turn on timer. At this time, the timer start to count from set value, and IR start to sink current driving IRLED transmitting. When Timer downward counter reach the duty value set at begin, IR cancel sinking current status at once. And the timer continues to count down until 0. Then IR start to sink current again, the timer reloads the set value again and count down. The carrier will generate automatically through IR pin with specific duty.

The equation of calculating the timer value to generate specific frequency carrier is shown below.

TM=(Ftcs/Fca)-1

TM: TM value in decimal

Ftcs: Frequency of timer clock

Fca: Frequency of carrier

Reference to Figure 3-1, RPT1[3:0]=0XD, RPT0[3:0]=0x2; DTHN[3:0]=0x8, DTLN[3:0]=0xC. The counter of TM[7:0]=0xD2=210, the counter of no driving {DTHN[3:0], DTLN[3:0]}=0x8C=140. The counter counts down to 0x08C, the IR turn over, then the counter counts down until to 0, the IR turn over again, it will get the one-third period of IR carrier frequency (one-third driving IRLED, two-third no driving). If 8MHz is

selected through TCS instruction as Timer source clock, the IR carrier frequency will be 37.915KHz ((210+1)*125ns=26375ns).

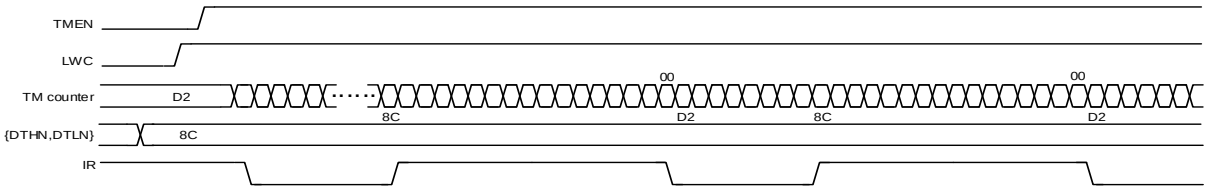


Figure 3-2 TX with Carrier diagram

In order to generate IR with no carrier, user need to set LWC = 0 at first, and turn on/off IR driving low through TXLD. The Figure 3-2 TX with no Carrier diagram shows the control sequence.

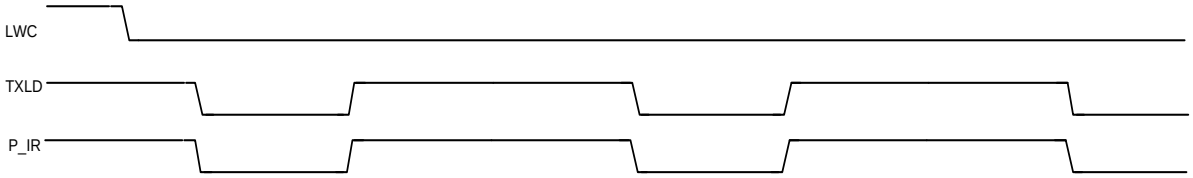


Figure 3-3 TX with no Carrier diagram

Example 3-2 Generate 38KHz (about 1/3 duty) carrier wave from IR

```
L_TST_IR:
    MVLA    0X3             ; Change Timer clock source to 8M
    CHTCS

    MVLA    0x8
    MVAT    DTHN            ; Setting timer duty high 4bits
    MVLA    0xB
    MVAT    DTLN            ; Setting timer duty low 4bits
    MVLR    0xD, RPT1       ; Setting Timer Data middle 4bits
    MVLR    0x1, RPT0       ; Setting Timer Data low 4bits
    LDTM                    ; Load rpt value to timer counter to generate 38K carrier

    MVLA    0x2
    MVAT    IRTX            ; Setting TXDT=1 With carrier
    MVLA    0x8
    MVAT    TMC             ; TM ON & 38K PWM signal output from IR pin
```

Example 3-3 Timer overflow

```
L_TST_BT_INT_TOF:
    MVLA    0X3             ; Change Timer clock source to 1M
    CHTCS
    MVLR    0xF, RPT1
    MVLR    0xF, RPT0
    LDTM                    ; Set timer value

    MVTA    TMC
    ORL     0x8
    MVAT    TMC             ; Set timer on
    MVLA    0x4
    MVAT    INT
    MVLA    0x7
    MVAT    INT             ; Set int on, detect timer ovf in interrupt process
```

Note: Carrier 1/3 duty means 1/3 period driving low, 2/3 period without driving.

3.8.3 TMC

The TMC register includes the status of TOF, and the switch for TM. Users can clear TOF by setting 0 to TMC[0]. The TM should be turned on if the IR application is applied.

Category	Bit	Description
TMC	[0]	TOF: Write0(clear)only; TM ON to activate TO Flag
	[1]	Reserved
	[2]	Reserved
	[3]	TMEN: TM will keep last value if ON after OFF without LDTM

3.8.4 IRTX

The TX/RX (Transmitter/Receiver) control diagram is shown in Figure 3-3. For TX application, the duty of IR output carrier is adjustable and the frequency can be defined by register TCS and TM. The IR large sink current can be set through relative option (I_{max}, 0.5*I_{max}, GPIO reverse output can be selected).

Category	Bit	Description
IRTX	[0]	TXLD: TX latch data, Write 1 to OFF TX
	[1]	LWC: Data L WO/W carrier,1: with carrier0: without carrier

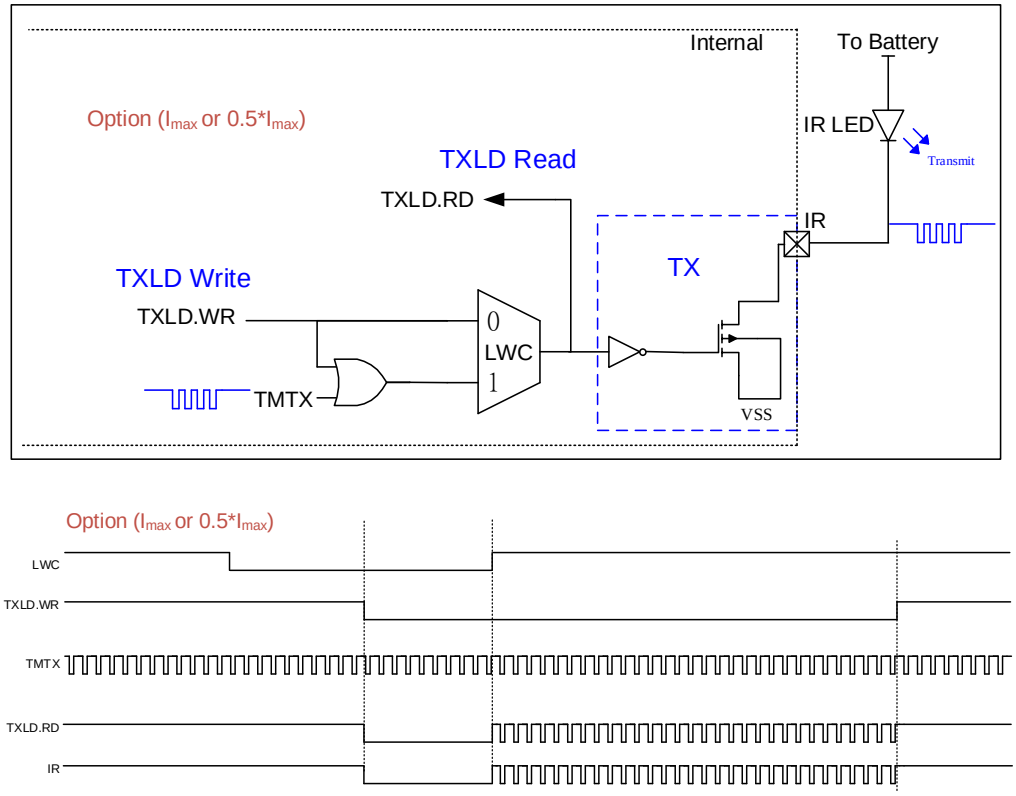


Figure 3-4 TX control & signal phase diagram of option (I_{max} or 0.5*I_{max})

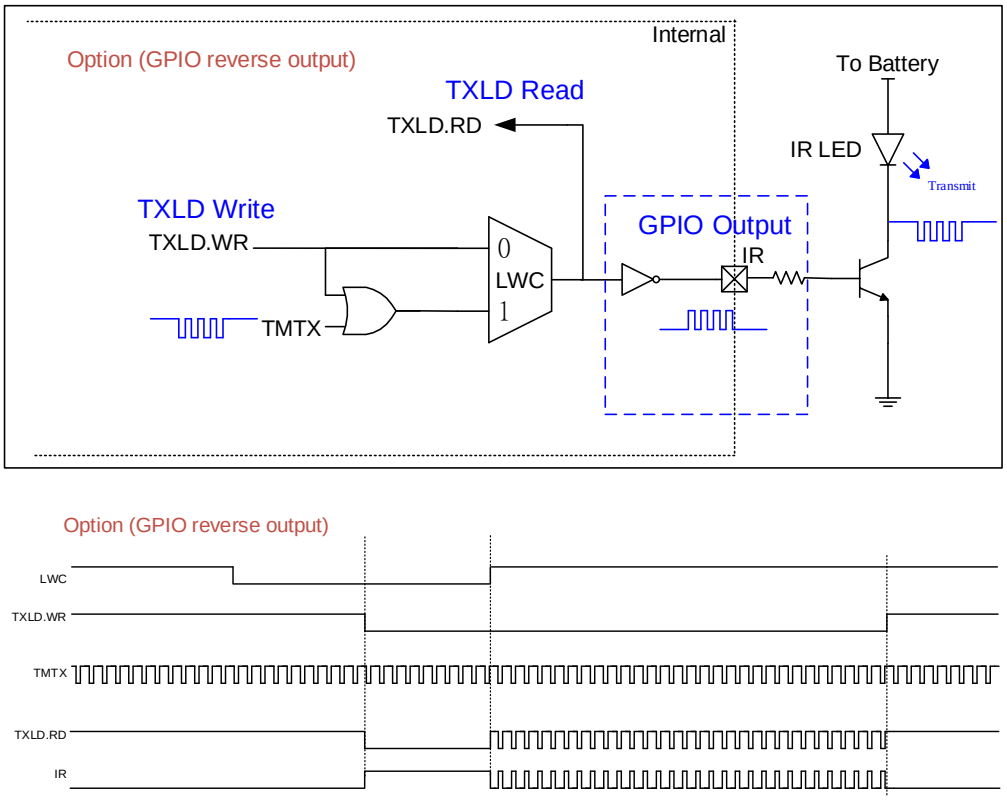


Figure 3-5 TX control & signal phase diagram of option (GPIO reverse output)

Example 3-4 Drive TX with no carrier

```
MVLA    0X1
MVAT    IRTX            ; No carrier, IR disable & keep floating
MVLA    0X0
MVAT    IRTX            ; No carrier, IR=0, driving large current to ground
```

3.9 Power Saving Mode

The relationship between power saving mode, reset & normal mode is shown below.

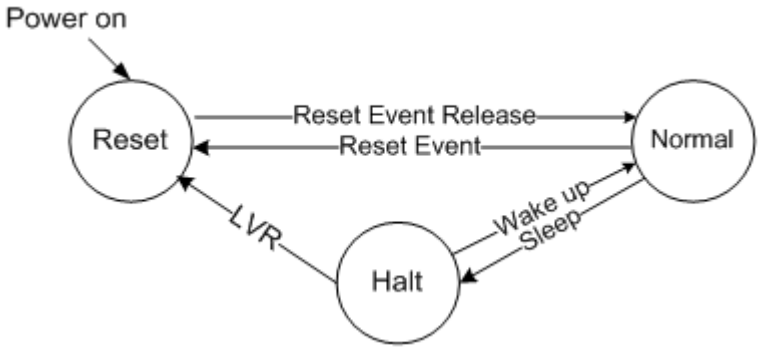


Figure 3-6 Power Saving Mode Flow Diagram

3.9.1 Halt Mode

The system enters the halt mode if the HALT command executed. The halt mode is also known as the sleep mode. As implied by the name, the IC falls asleep and the system clock is completely turned off, so all the IC functions are halted and it minimizes the power consumption.

The only way to wake-up the sleeping system is an input port wake-up. The IC keeps monitoring the input pads during the halt mode. If the input status of any input pad changes to low, the system will be woken-up. Then the succeeding instructions after the HALT instruction will be executed after the wake-up stable time (about 60us). So before executing the HALT instruction, users have to keep in mind that the input port status is high.

If the IC is waked-up from the halt mode by the occurrence of LVR, it goes into the reset procedure.

Example 3-5 Halt Mode operation

```
L_HLT_Loop:
        MVLA    0xE
        MVAT    INT                     ; Clear watch dog

        MVTA    PA
        CPAL    0xF
        JMP     L_HLT_Loop
        MVTA    PB
        CPAL    0xF
        JMP     L_HLT_Loop
        MVTA    PC
        CPAL    0xF
        JMP     L_HLT_Loop
        MVTA    PD
        CPAL    0xF
        JMP     L_HLT_Loop              ; Confirm I/O is high status if I/O is set as input status
```

```
L_HLT_Proc:
        NOP
        NOP
        NOP
        HALT                            ; Enter Halt mode
        NOP                             ; If input status I/O change to low, wake up
        NOP
        NOP
        JMP     L_HLT_Loop
```

3.9.2 T-type Scan Mode

In T-type scanning application, each port (PA~PD) can be selected as scan key independently by option PXx (X=A~C, x=0~3) & PD0. And setting the port as bi-direction input ports is necessary under T-type scan mode. It works as input with pull-high resistor and output fixed frequency low pulse in halt mode. Any of the keys touch would cause system wake-up. Meanwhile, the frequency of key scan can be adjusted by option codes, such as about 15.625Hz, 31.25Hz, 62.5Hz and 125Hz.

Chapter 4. Instruction Set

4.1 Instruction Classified Table

Item	Inst.	Op1	Op2	Operation	Inst. Length	Exec. Cycle	Oper. Flag	Flag Affected
Arithmetic Instructions								
1	INCM	6m		M = M +1	1	1		C, Z
2	DECM	6m		M = M - 1	1	1		C, Z
3	ADDM	6m		M = A + M + C	1	1	C	C, Z
4	XORM	6m		M = A ^ M	1	1		Z
5	ANDM	6m		M = A & M	1	1		Z
6	ORM	6m		M = A | M	1	1		Z
7	MVAM	6m		Move A to M	1	1		
8	MVMA	6m		Move M to A	1	1		Z
9	MVRM	4m	2r	Move R to M	1	1		
10	MVMR	4m	2r	Move M to R	1	1		
11	MVLR	4L	2r	Move L to R	1	1		
12	ADDL	4L		A = A + L + C	1	1	C	C, Z
13	XORL	4L		A = A ^ L	1	1		Z
14	ANDL	4L		A = A & L	1	1		Z
15	ORL	4L		A = A | L	1	1		Z
16	MVLA	4L		Move L to A	1	1		
17	MVAT	4t		Move A to T	1	1		
18	MVTA	4t		Move T to A	1	1		Z
19	RSTC			Reset C = 0	1	1		C
20	SETC			Set C = 1	1	1		C
21	INCA			A = A + 1	1	1		C, Z
22	DECA			A = A - 1	1	1		C, Z
Conditional Instructions								
23	CPAM	6m		Skip if A = M	1	1		
24	CPMZ	6m		Skip M = 0x0	1	1		
25	CPAL	4L		Skip if A = L	1	1		
26	CPCZ			Skip if C = 0	1	1	C	
27	CPZZ			Skip if Z = 0	1	1	Z	
IR Instructions								
28	CHTCS			Move A[1:0] to TCS	1	1		
29	LDTM			Move RPT to TM	1	1		
30	RBTM			Move TM to RPT	1	1		
Other Instructions								
31	RBRO	1n		Move ROM[RPT] data to ROD and ACC	1	3		
32	JMP	14a		Jump to Address	2	2		
33	CALL	14a		Jump to Address, and Move PC+2 to RPT	2	2		
34	IRET			Move STK to PC	2	2		
35	LDPC			Move RPT to PC	1	2		

Item	Inst.	Op1	Op2	Operation	Inst. Length	Exec. Cycle	Oper. Flag	Flag Affected
36	RBPC			Move PC+1 to RPT	1	2		
37	HALT			Enter HALT mode	1	1		
38	NOP			No operation	1	1		

A : 4-bit Accumulator data

C : 1-bit carry flag data

M : 4-bit RAM or memory register data

R : 4-bit memory register data

L : 4-bit immediately literal data

T : 4-bit System register data

Z : 1-bit zero flag data

TCS : 2-bit clock source selection register of IR carrier frequency

RPT : Multi-function register data

TM : 16-bit timer data of IR carrier

ROM : 10-bit ROM data

ROD : ROM data access register data

PC : Program counter address pointer

STK : Interrupt dedicated stack address pointer

a : ROM address

m : RAM or memory register address

n : data address Plus/Not plus 1

p : RAM page

r : Memory register address

t : System register address

4.2 Instruction Descriptions

4.2.1 Arithmetic Instructions

INCM m

Function: Add 1 to M of address m, and save the result back to M.

Operation: M ← M + 1

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C, Z

Example: INCM m0

Before Instruction

M0=0x0

After Instruction

M0=0x1, C=0, Z=0

DECM m

Function: Subtract 1 from M of address m, and save the result back to M.

Operation: M ← M - 1

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C, Z

Example: DECM m0

Before Instruction

M0=0x0

After Instruction

M0=0xF, C=0, Z=0

ADDM m

Function: Add M of address m and C to A, and save the result back to M.

Operation: M ← A + M + C

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1

Operative Flags: C

Flags Affected: C, Z

Example: ADDM m0

Before Instruction

A=0x7, M0=0xA, C=0

After Instruction

A=0x7, M0=0x1, C=1, Z=0

XORM m

Function: Exclusive OR A with M of address m, and the result is save back to M.

Operation: M ← A ^ M

Operand: 0x0≦m≦3F

Words: 1

Cycles: 1

Operative Flags:None

Flags Affected: Z

Example: XORM m0

Before Instruction

A=0x3, M0=0xB

After Instruction

A=0x3, M0=0x8, Z=0

ANDM m

Function: AND A with M of address m, and save the result back to M.

Operation: M ← A & M

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: ANDM m0

Before Instruction

A=0x7, M0=0xA

After Instruction

A=0x7, M0=0x2, Z=0

ORM m

Function: Inclusive OR A with M of address m, and save the result back to M.

Operation: M ← A | M

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: ORM m0

Before Instruction

A=0x3, M0=0x8

After Instruction

A=0x3, M0=0xB, Z=0

MVAM m

Function: Move A to M of address m.

Operation: M ← A

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVAM m0

Before Instruction

A=0x8

After Instruction

M0=0x8

MVMA m

Function: Move M of address m to A.

Operation: A ← M

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: MVMA m0

Before Instruction

M0=0x8

After Instruction

A=0x8

MVRM m, r

Function: Move R to M. Which R address MSB is 0 and M address MSB 2-bit is 0x3.

Operation: M ← R

Operand: 0x0≦m≦0xF

0x0≦r≦0x3

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVRM m0, 0x0

Before Instruction

PG=0, RPT0=0x8

After Instruction

M30=0x8

MVMR m, r

Function: Move M to R. Which R address MSB is 0 and M address MSB 2-bit is 0x3.

Operation: R ← M

Operand: 0x0≦m≦0xF

0x0≦r≦0x3

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVMR m1, 0x1

Before Instruction

PG=0, M31=0x8

After Instruction

RPT1=0x8

MVLR L, r

Function: Move the immediate constant value to R. Which R address MSB is 0.

Operation: R ← L

Operand: 0x0≦L≦0xF

0x0≦r≦0x3

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVLR 0x7, 0x3

After Instruction

RPT3=0x7

ADDL L

Function: Add L and C to A.

Operation: A ← A + L+ C

Operand: 0x0≦L≦0xF

Words: 1

Cycles: 1

Operative Flags: C

Flags Affected: C, Z

Example: ADDL 0x9

Before Instruction

A=0xB, C=1

After Instruction

A=0x5, C=1, Z=0

XORL L

Function: Exclusive OR A with L.

Operation: A ← A ^ L

Operand: 0x0≦L≦0xF

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: XORL 0xA

Before Instruction

A=0x9

After Instruction

A=0x3, Z=0

ANDL L

Function: AND A with L.

Operation: A ← A & L

Operand: 0x0≦L≦0xF

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: ANDL 0xA

Before Instruction

A=0x0

After Instruction

A=0x0, Z=1

ORL L

Function: Inclusive OR AL.

Operation: A ← A | L

Operand: 0x0≦L≦0xF

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: ORL 0xA

Before Instruction

A=0x9

After Instruction

A=0xB, Z=0

MVLA L

Function: Move L to A.

Operation: A ← L

Operand: 0x0≦L≦0xF

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVLA 0x7

After Instruction

A=0x7

MVAT t

Function: Move A to T of address t.

Operation: T ← A

Operand: 0x0≦t≦0xF

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVAT 0x0

Before Instruction

A=0x8

After Instruction

VOL=0x8

MVTA t

Function: Move T of address t to A.

Operation: A ← T

Operand: 0x0≦t≦0xF

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: Z

Example: MVTA 0x8

Before Instruction

Port A data = 0x4

After Instruction

A=0x4

RSTC

Function: Clear C to 0.

Operation: C ← 0

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C

Example: RSTC

Before Instruction

C=1

After Instruction

C=0

SETC

Function: Set C to 1.

Operation: C ← 1

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C

Example: SETC

Before Instruction

C=0

After Instruction

C=1

INCA

Function: Add 1 to A.

Operation: A ← A + 1

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C, Z

Example: INCA

Before Instruction

A=0xF

After Instruction

A=0x0, C=1, Z=1

DECA

Function: Subtract 1 from A.

Operation: A ← A - 1

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: C, Z

Example: DECA

Before Instruction

A=0x1

After Instruction

A=0x0, C=1, Z=1

4.2.2 Conditional Instructions

CPAM m

Function: Skip the next instruction if A equals M of address m.

Operation: Skip next if A=M

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1, (2, 3)

Operative Flags: None

Flags Affected: None

Example: CPAM m0

Inst1

Inst2

After Instruction

If A≠M0, `Inst1' is executed.

If A=M0, `Inst1' is discarded, and `Inst2' is executed.

CPMZ m

Function: Skip the next instruction if M of address m equals zero.

Operation: Skip next if M=0x0

Operand: 0x0≦m≦0x3F

Words: 1

Cycles: 1, (2, 3)

Operative Flags: None

Flags Affected: None

Example: CPMZ m0

Inst1

Inst2

After Instruction

If M0≠0x0, `Inst1' is executed.

If M0=0x0, `Inst1' is discarded, and `Inst2' is executed.

CPAL L

Function: Skip the next instruction if A equals L.

Operation: Skip next if A=L

Operand: 0x0≦L≦0xF

Words: 1

Cycles: 1, (2, 3)

Operative Flags: None

Flags Affected: None

Example: CPAL 0x4

CALL a1

CALL a2

After Instruction

If A≠0x4 `CALL a1' is executed

If A=0x4 `CALL a1' is discarded, and `CALL a2' is executed

CPCZ

Function: Skip the next instruction if C equals zero.

Operation: Skip next if C=0

Operand: None

Words: 1

Cycles: 1, (2, 3)

Operative Flags: C

Flags Affected: None

Example: CPCZ

CALL a1

CALL a2

After Instruction

If C≠0 `CALL a1' is executed

If C=0 `CALL a1' is discarded, and `CALL a2' is executed

CPZZ

Function: Skip the next instruction if Z equals zero.

Operation: Skip next if Z=0

Operand: None

Words: 1

Cycles: 1, (2, 3)

Operative Flags: Z

Flags Affected: None

Example: CPZZ

CALL a1

CALL a2

After Instruction

If Z≠0 `CALL a1' is executed

If Z=0 `CALL a1' is discarded, and `CALL a2' is executed

4.2.3 IR Instructions

CHTCS

Function: Move A[1:0] to TCS of IR carrier timer.

Operation: TCS ← A[1:0]

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVLA 0x03

CHTCS

Before Instruction

A=0x0, TCS=0x0

After Instruction

A=0x3, TCS=0x3

LDTM

Function: Load RPT[7:0] value to TM of IR carrier.

Operation: TM ← RPT[7:0]

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: MVLR 0x2, 0x1

MVLR 0x1, 0x0

LDTM

Before Instruction

RPT=0x21, TM=0x00

After Instruction

RPT=0x21, TM=0x21

RBTM

Function: Move TM counter to RPT[7:0].

Operation: RPT[7:0] ← TM

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: RBTM

Before Instruction

RPT = 0x55

TM = 0xAA

After Instruction

RPT =0xAA

4.2.4 Other Instructions

JMP a

Function: Unconditionally jump by a direct address a.

Operation: PC ← a

Operand: 0x0≦a≦0x2FF

Words: 2

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: JMP a1

Before Instruction

PC=a0

After Instruction

PC=a1

CALL a

Function: Call subroutine by a direct address a, and save next address to RPT.

Operation: RPT ← PC+2

PC ← a

Operand: 0x0≦a≦0x2FF

Words: 2

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: CALL a1

Before Instruction

PC=a0

After Instruction

PC=a1, RPT=a0+2

RBRO n

Function: Read ROM data out to A and ROD using the RPT as address(data pointer).

Operation: A ← ROM data [3:0]

ROD1 ← ROM data [7:4]

ROD2 ← ROM data [9:8]

RPT ← RPT + n

Operand: 0≦n≦1

Words: 1

Cycles: 3

Operative Flags: None

Flags Affected: None

Example: RBRO 1

After Instruction

A=ROM[3:0] @ RPT

ROD1=ROM[7:4] @ RPT

ROD2=ROM[9:8] @ RPT

RPT=RPT+1

IRET

Function: Return from the interrupt sub-routine.

Operation: PC ← STK

Operand: None

Words: 2

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: IRET

Before Instruction

PC=a0

After Instruction

PC=STK

ACC, PG, C, and Z are restored to the values that are backed up when entering the ISR.

RBPC

Function: Read address in PC to RPT.

Operation: RPT ← PC+1

Operand: None

Words: 1

Cycles: 2

Operative Flags: None

Flags Affected: None

Example:RBPC

Before Instruction

PC=0x234

After Instruction

RPT=0x235

HALT

Function: Enter the halt (sleep) mode.

Operation: Stop system clock

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: HALT

After Instruction

The system enters the halt mode and the system clock is halted.

LDPC

Function: Load RPT to PC. Unconditionally jump by the indirect address RPT. The address should be loaded into RPT first.

Operation: PC ← RPT

Operand: None

Words: 1

Cycles: 2

Operative Flags: None

Flags Affected: None

Example: LDPC

Before Instruction

RPT=0x231

After Instruction

PC=0x231

NOP

Function: No operation.

Operation: None

Operand: None

Words: 1

Cycles: 1

Operative Flags: None

Flags Affected: None

Example: NOP

After Instruction

No operation for 1 cycle.