# NT68P81

## USB Keyboard Micro-Controller

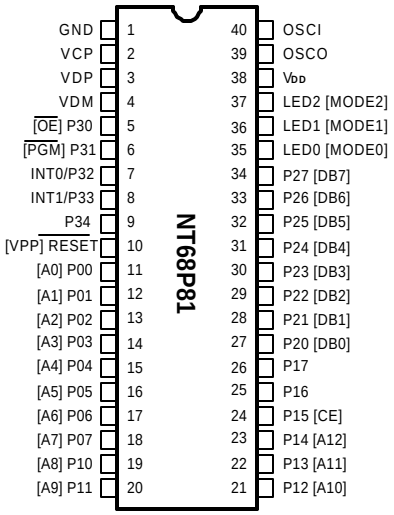
### Features

- Built-in 6502C 8-bit CPU
- 3 MHz CPU operation frequency when oscillator is running at 6 MHz
- 6K bytes of OTP (one time programming) ROM
- 256 bytes of SRAM
- One 8-bit programmable base timer with pre-divider circuit
- 29 programmable bi-directional I/O pins including two external interrupts
- 3 LED direct sink pins with internal serial resistors
- On-chip oscillator (Crystal or Ceramic Resonator)
- Watch-dog timer reset
- Built-in power-on reset
- USB interface
- 3 supported endpoints
- Remote wakeup provided
- CMOS technology for low power consumption
- 40-pin DIP package, 42-pad Dice form and COB

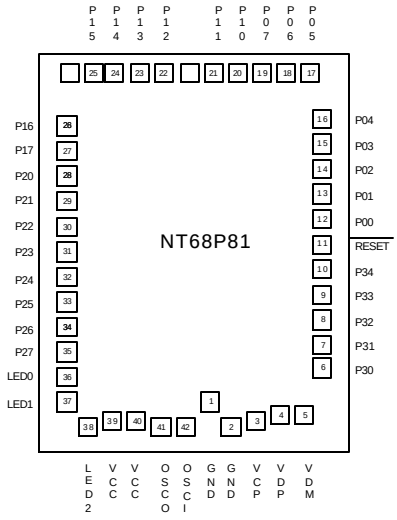
### General Description

The NT68P81 is a single chip micro-controller for USB keyboard applications. It incorporates a 6502C 8-bit CPU core, 6K bytes of OTP ROM, and 256 bytes of RAM used as working RAM and stack area. It also includes 29 programmable bi-directional I/O pins with built-in resistors, and one 8-bit pre-loadable base timer. Additionally, it includes a built-in power-on reset, a built-in low voltage reset, an oscillator that requires crystal or ceramic resonator applied, and a watch-dog timer that prevents system standstill.

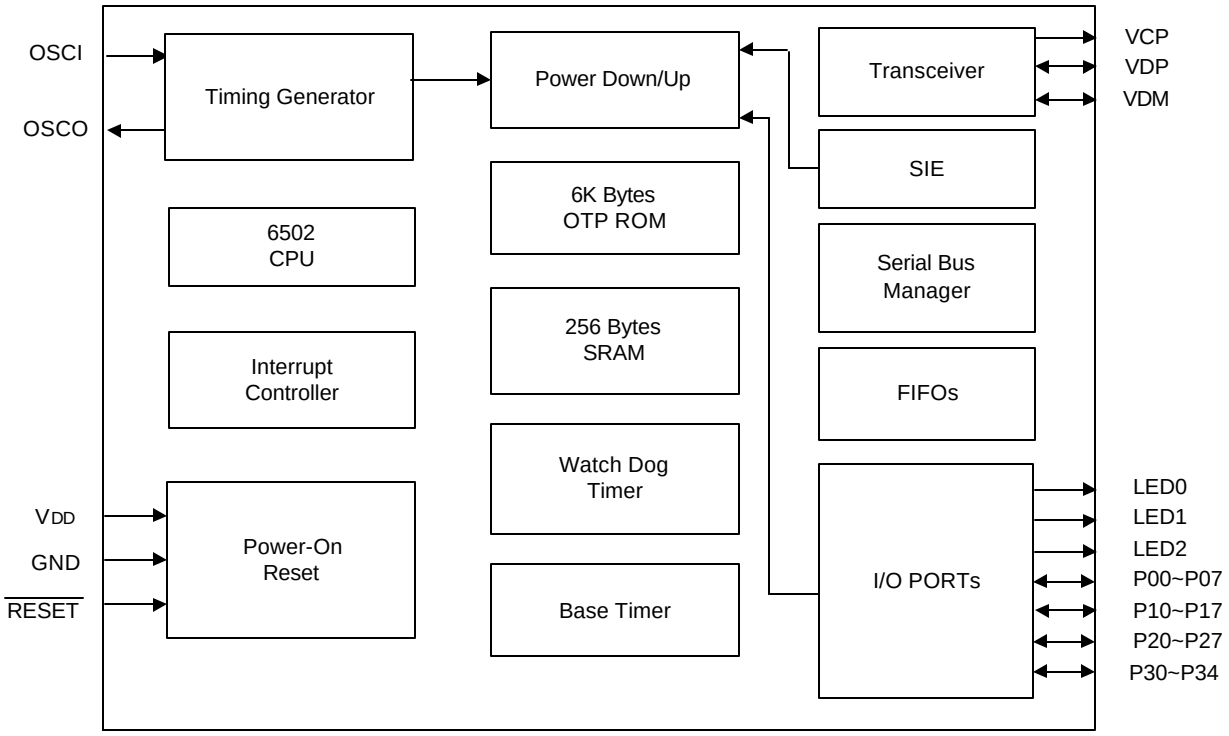
### Pin Configuration




### Pad Configuration



## Block Diagram



## Pin and Pad Descriptions

| Pin No. | Pad No. | Designation | I/O | Shared with OTP[I/O] | Description |
|---|---|---|---|---|---|
| 1 | 1,2 | GND | P | | Ground |
| 2 | 3 | VCP | O | | USB 3.3V driver |
| 3 | 4 | VDP | I/O | | USB data plus |
| 4 | 5 | VDM | I/O | | USB data minus |
| 5 | 6 | P30 | I/O | | Bi-directional I/O pin |
| | | | | $\overline{OE}$ [I] | Program output enable |
| 6 | 7 | P31 | I/O | | Bi-directional I/O pin |
| | | | | $\overline{PGM}$ [I] | Program control |
| 7 | 8 | P32/INT0 | I/O | | Bi-directional I/O shared with INT0 |
| 8 | 9 | P33/INT1 | I/O | | Bi-directional I/O shared with INT1 |
| 9 | 10 | P34 | I/O | | Bi-directional I/O pin |
| 10 | 11 | $\overline{RESET}$ | I | | Internally pulled down resistor |
| | | | | VPP [P] | Program supply voltage |
| 11 ~ 18 | 12 ~ 19 | P00 ~ P07 | I/O | | Bi-directional I/O pin |
| | | | | A0 ~ A7 [I] | Program address buffer |
| 19 ~ 23 | 20 ~ 24 | P10 ~ P14 | I/O | | Bi-directional I/O pin |
| | | | | A8 ~ A12 | Program address buffer |
| 24 | 25 | P15 | I/O | | Bi-directional I/O pin |
| | | | | CE [I] | Program chip enable |
| 25 | 26 | P16 | I/O | | Bi-directional I/O pin |
| | | | | VPIH[I] | OTP Program Input Voltage High |
| 26 | 27 | P17 | I/O | | Bi-directional I/O pin |
| 27 ~ 34 | 28 ~ 35 | P20 ~ P27 | I/O | | Bi-directional I/O pin |
| | | | | DB0 ~ DB7 [I/O] | Program data buffer |
| 35 | 36 | LED0 | O | | LED direct sink |
| | | | | MODE0 [I] | Mode selection |
| 36 | 37 | LED1 | O | | LED direct sink |
| | | | | MODE1 [I] | Mode selection |
| 37 | 38 | LED2 | O | | LED direct sink |
| | | | | MODE2 [I] | Mode selection |
| 38 | 39,40 | $V_{DD}$ | P | | Power supply (+5V) |
| 39 | 41 | OSCO | O | | Crystal oscillator output |
| | | | | CLK[I] | Program Clock |

| | | | | | |
|---|---|---|---|---|---|
| 40 | 42 | OSCI | I | | Crystal oscillator input |
| | | | | VPIL[I] | OTP Program Input Voltage Low |

* [ ]: OTP Mode

## Functional Description

### 1. 6502C CPU

The 6502C is an 8-bit CPU that provides 56 instructions, decimal and binary arithmetic, thirteen addressing modes, true indexing capability, programmable stack pointer and variable length stack, a wide selection of addressable memory range, and an interrupt input. Other features are also included.
The CPU clock cycle is 3MHz (6MHz system clock divided by 2). Please refer to 6502 data sheet for more detailed information.

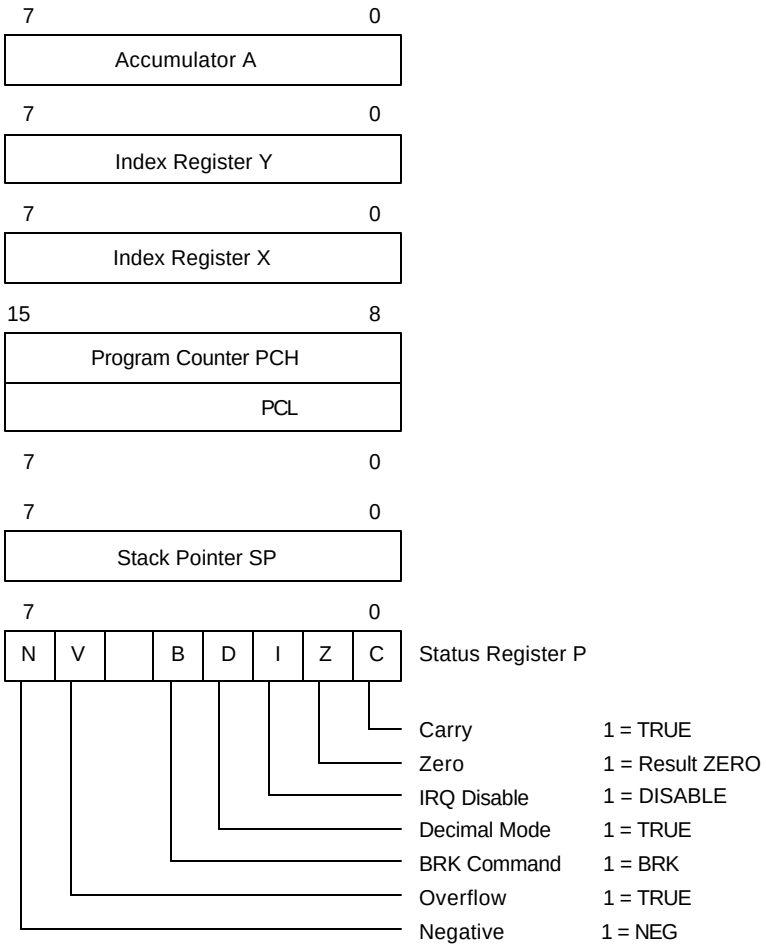



**Figure 1. 6502 CPU Registers and Status Flags**

**2.Instruction Set List**

| Instruction Code | Meaning | Operation |
|---|---|---|
| ADC | Add with carry | A + M + C → A, C |
| AND | Logical AND | A · M → A |
| ASL | Shift left one bit | C ← M7 · · · M0 ← 0 |
| BCC | Branch if carry clear | Branch on C=0 |
| BCS | Branch if carry set | Branch on C=1 |
| BEQ | Branch if equal to zero | Branch on Z=1 |
| BIT | Bit test | A · M, M7 → N, M6 → V |
| BMI | Branch if minus | Branch on N=1 |
| BNE | Branch if not equal to zero | Branch on Z=0 |
| BPL | Branch if plus | Branch on N=0 |
| BRK | Break | Forced interrupt PC + 2↓ PC↓ |
| BVC | Branch if overflow clear | Branch on V=0 |
| BVS | Branch if overflow set | Branch on V=1 |
| CLC | Clear carry | 0 → C |
| CLD | Clear decimal mode | 0 → D |
| CLI | Clear interrupt disable bit | 0 → I |
| CLV | Clear overflow | 0 → V |
| CMP | Compare accumulator to memory | A − M |
| CPX | Compare with index register X | X − M |
| CPY | Compare with index register Y | Y − M |
| DEC | Decrement memory by one | M − 1 → M |
| DEX | Decrement index X by one | X − 1 → X |
| DEY | Decrement index Y by one | Y − 1 → Y |
| EOR | Logical exclusive-OR | A ⊕ M→A |
| INC | Increment memory by one | M + 1 → M |
| INX | Increment index X by one | X + 1 → X |
| INY | Increment index Y by one | Y + 1 → Y |
| JMP | Jump to new location | (PC + 1) → PCL, (PC + 2) → PCH |
| JSR | Jump to subroutine | PC + 2↓, (PC + 1) → PCL, (PC + 2) → PCH |

**Instruction Set List (contiuned)**

| Instruction Code | Meaning | Operation |
|---|---|---|
| LDA | Load accumulator with memory | M → A |
| LDX | Load index register X with memory | M → X |
| LDY | Load index register Y with memory | M → Y |
| LSR | Shift right one bit | 0 → M7・・・M0 → C |
| NOP | No operation | No operation (2 cycles) |
| ORA | Logical OR | A + M → A |
| PHA | Push accumulator on stack | A ↓ |
| PHP | Push status register on stack | P ↓ |
| PLA | Pull accumulator from stack | A ↑ |
| PLP | Pull status register from stack | P ↑ |
| ROL | Rotate left through carry | C ← M7・・・M0 ← C |
| ROR | Rotate right through carry | C → M7・・・M0 → C |
| RTI | Return from interrupt | P ↑, PC ↑ |
| RTS | Return from subroutine | PC ↑, PC+1 → PC |
| SBC | Subtract with borrow | A − M − C → A, C |
| SEC | Set carry | 1 → C |
| SED | Set decimal mode | 1 → D |
| SEI | Set interrupt disable status | 1 → I |
| STA | Store accumulator in memory | A → M |
| STX | Store index register X in memory | X → M |
| STY | Store index register Y in memory | Y → M |
| TAX | Transfer accumulator to index X | A → X |
| TAY | Transfer accumulator to index Y | A → Y |
| TSX | Transfer stack pointer to index X | S → X |
| TXA | Transfer index X to accumulator | X → A |
| TXS | Transfer index X to stack pointer | X → S |
| TYA | Transfer index Y to accumulator | Y → A |

*For more detailed specifications, please refer to 6502 programming data book.

### 3. OTP ROM: 6K X 8 bits

The built-in OTP ROM program code, executed by the 6502 CPU, has a capacity of 6K x 8-bit and is addressed from E800H to FFFFH. It can be programmed by the universal EPROM writer through a conversion adapter and programming configuration such as INTEL - 27C64. In the OPERATING mode, the OTP ROM is integrated with the system and it cannot be directly accessed. When the user wants to work with the OTP ROM alone, the user must first enter the PROGRAMMING mode by setting: PIN < $\overline{\text{RESET}}$ = VPP>. At this time, through multiplex pins, we can use familiar procedures to program and verify the OTP ROM block with the universal programmer.

**OTP ROM Mega Cell D.C. Electrical Characteristics (READ Mode)**

($V_{DD}$ = 5V, $T_A$ = 25℃, unless otherwise specified)

| Symbol | Parameter | Min. | Typ. | Max. | Unit | Test Conditions | Note |
|---|---|---|---|---|---|---|---|
| $V_{IH}$ | Input Voltage | $V_{DD}$ - 0.3 | | $V_{DD}$ + 0.3 | V | | 1 |
| $V_{IL}$ | | -0.3 | | 0.3 | V | | 1 |
| $I_{IL}$ | Input Current | | | +/-10 | µA | | |
| $I_{OH}$ | Output Voltage | -400 | | | µA | $V_{DD}$ = 5V, $V_{OH}$ = 4.5V | |
| $I_{OL}$ | | 1 | | | mA | $V_{DD}$ = 5V, $V_{OL}$ = 0.5V | |
| $I_{DD}$ | Operating Current | | | 1 | mA | F = 3MHz | 2 |
| $I_{STB1}$ | Standby Current | | | 100 | µA | | 3 |

Note: 1. All inputs and outputs are CMOS compatible
2. F = 3MHz, $I_{out}$ = 0mA, CE = $V_{IH}$. $V_{DD}$ = 5V
3. CE = $V_{IH}$, $\overline{\text{OE}}$ = $V_{IL}$, $V_{DD}$ = 5V

**OTP ROM Mega Cell A.C. Electrical Characteristics (READ Mode)**

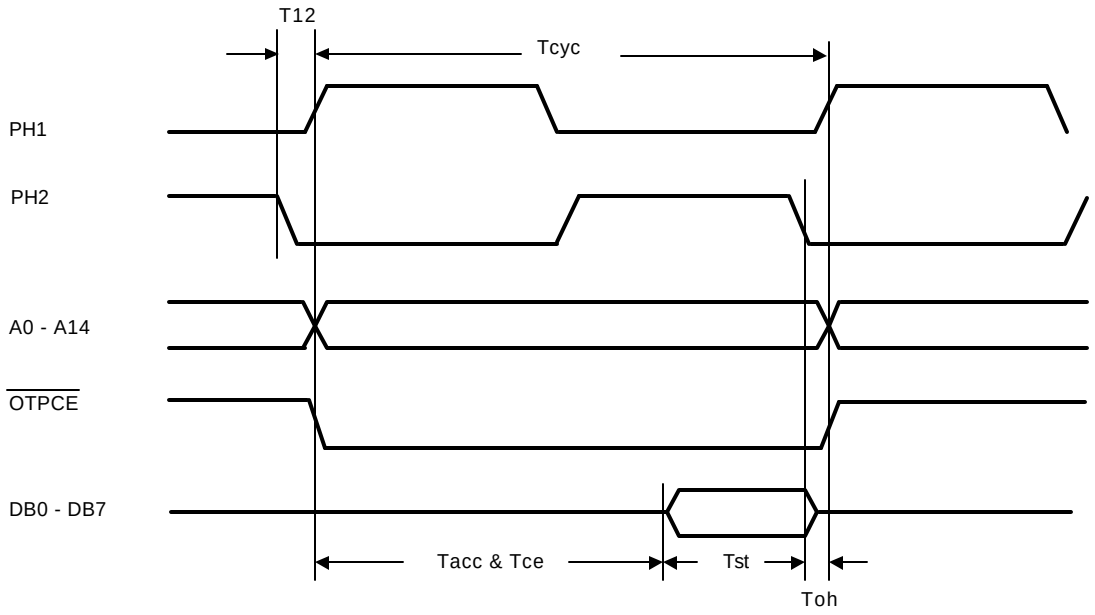
($V_{DD}$ = 5V, $T_A$ = 25℃, unless otherwise specified)

| Symbol | Parameter | Min. | Max. | Unit | Conditions |
|---|---|---|---|---|---|
| $T_{cyc}$ | Cycle Time | 250 | | ns | 4.5V < $V_{DD}$ < 5.5V |
| $T_{12}$ | Non-overlap Time to PH1 & PH2 | 5 | 65 | ns | |
| $T_{acc}$ | Address Access Time | | 145 | ns | |
| $T_{ce}$ | OTPCE to Output Valid | | 145 | ns | |
| $T_{st}$ | Output Data Setup Time | 20 | | ns | |
| $T_{oh}$ | Output Data Hold Time | 0 | | ns | |

**OTP ROM Mega Cell A.C. Test Conditions**

| | |
|---|---|
| Output Load | 1 CMOS Gate and CL = 10pF |
| Input Pulse Rise and Fall Times | 10ns Max. |
| Input Pulse Levels | 0V to 5V |
| Timing Measurement Reference Level | Inputs 0V and 5V Outputs 0.3V and 4.7V |

**OTP ROM Mega Cell Timing Waveforms (READ mode)**




**OTP ROM Mega Cell D.C. Electrical Characteristics (PROGRAMMING Mode)**

($V_{DD}$ = 5V, $T_A$ = 25℃, unless otherwise specified)

| Symbol | Parameter | Min. | Typ. | Max. | Unit | Test Conditions | Note |
|---|---|---|---|---|---|---|---|
| $V_{DD}$ | Supply Voltage | | 6 | 6.5 | V | | 4 |
| VPP | | 10.5 | | 12.75 | V | | |
| $V_{IH}$ | Input Voltage | 2 | | $V_{DD}$ + 0.3 | V | | |
| $V_{IL}$ | | -0.3 | | 0.6 | V | | |
| $I_{IL}$ | Output Current | | | +/-10 | μA | | |
| $I_{OH}$ | Output Current | -400 | | | μA | $V_{DD}$ = 5V, $V_{OH}$ = 4.5V | |
| $I_{OL}$ | | 1 | | | mA | $V_{DD}$ = 5V, $V_{OL}$ = 0.5V | |
| $I_{DD}$ | Operating Current | | | 30 | mA | | |
| $I_{PP}$ | | | | 20 | mA | VPP = 12.75V | |
| CLK | Input Clock | | 53.203424 | | MHz | | |
| VPIH | Input Voltage | 2 | | $V_{DD}$ + 0.3 | V | | |
| VPIL | | -0.3 | | 0.6 | V | | |

Note: 4. For reliability concerns, we suggest $V_{DD}$ = 6V & VPP = 12.75V for testing OTP ROM AC characteristics in PROGRAMMING mode, and the same condition is suggested for universal programmer supply voltage.

**OTP ROM Mega Cell A.C. Electrical Characteristics (PROGRAMMING Mode)**

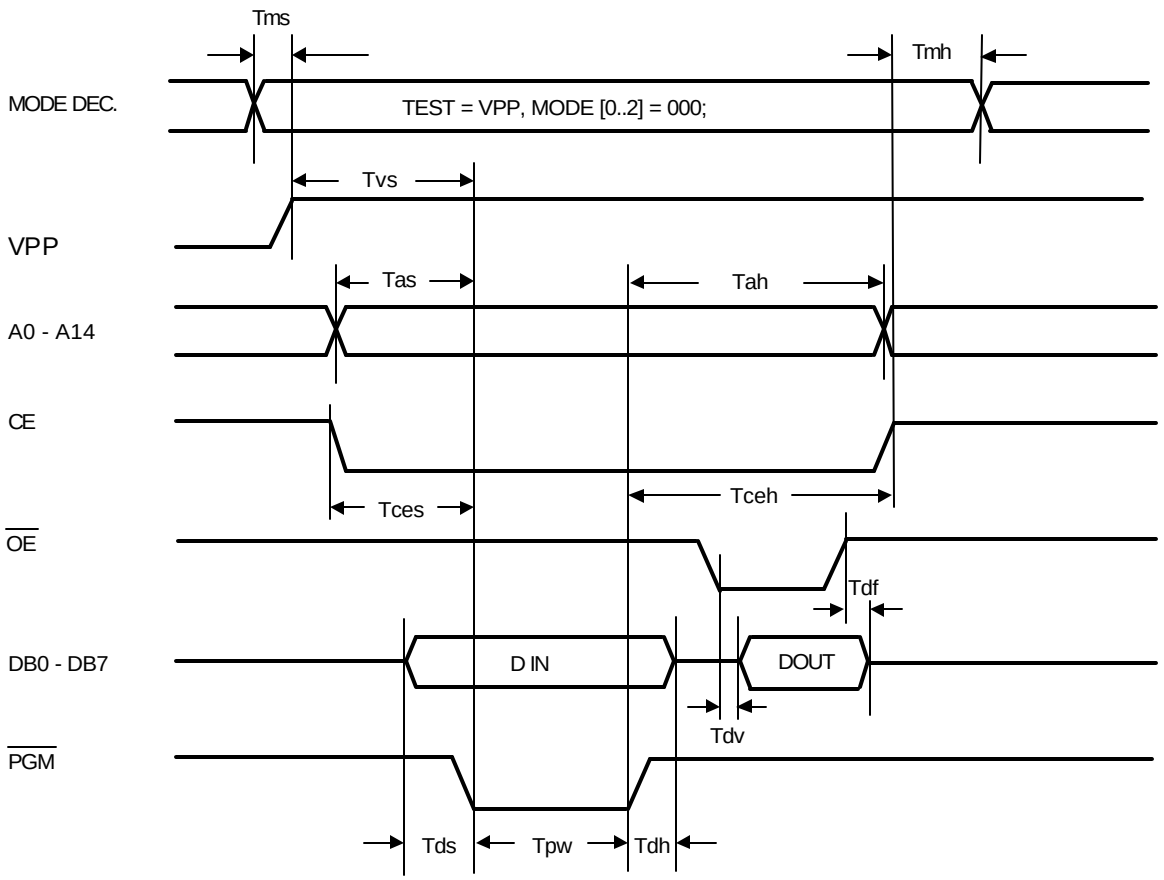
($T_A$ = 25°C, unless otherwise specified)

| Symbol | Parameter | Min. | Typ. | Max. | Unit | Test Conditions | Note |
|---|---|---|---|---|---|---|---|
| $T_{ms}$ | Mode Decode Setup Time | 2 | | | µs | | |
| $T_{mh}$ | Mode Decode Hold Time | 2 | | | µs | | |
| $T_{as}$ | Address Setup Time | 2 | | | µs | | |
| $T_{ah}$ | Address Hold Time | 2 | | | µs | | |
| $T_{ces}$ | CE Setup Time | 2 | | | µs | | |
| $T_{ceh}$ | CE Hold Time | 2 | | | µs | | |
| $T_{ds}$ | Date Setup Time | 2 | | | µs | | |
| $T_{dh}$ | Data Hold Time | 2 | | | µs | | |
| $T_{vs}$ | VPP Setup Time | 2 | | | µs | | |
| $T_{pw}$ | Program Pulse Width | | | 100 | µs | | |
| $T_{dv}$ | $\overline{OE}$ to Output Valid | | | 150 | ns | | |
| $T_{df}$ | $\overline{OE}$ to Output High-Z | | | 90 | ns | CE = $V_{IL}$ | |

**OTP ROM Mega Cell A.C. Test Conditions**

| | |
|---|---|
| Output Load | 1 TTL Gate and C L = 100pF |
| Input Pulse Rise and Fall Times | 10ns max. |
| Input Pulse Levels | 0.45V to 2.4V |
| Timing Measurement Reference Level | Inputs 0.8V and 2.2V<br>Outputs 0.8V and 2.4V |

**OTP ROM Mega Cell Timing Waveform (Program)**




Note: 5. $V_{DD}$ must be applied simultaneously or before VPP and cut off simultaneously or after VPP.

6. Removing the device from the socket or setting the device in socket with VPP = 12.75V may cause permanent damage.

**OTP ROM Mega Cell Mode Selection**

| $\overline{RESET}$ = 12.75V, VPIL = $V_{IL}$, VPIH = $V_{IH}$ | Mode [0..2] | Mode | CE | $\overline{OE}$ | VPP | DB0-DB7 |
|---|---|---|---|---|---|---|
| not VPP | — | Normal Operating | — | — | — | — |
| VPP | 000 | Output Disable | — | $V_{IH}$ | — | high-Z |
| VPP | 000 | Program | $V_{IH}$ | $V_{IH}$ | VPP | data in |
| VPP | 000 | Program Verify | $V_{IH}$ | $V_{IL}$ | — | data out |
| VPP | 000 | Program Inhibit (Standby) | $V_{IL}$ | — | VPP | high-Z |
| VPP | 001 | Security (Program) | $V_{IH}$ | — | VPP | data in |
| VPP | 010 | Word-line Stress | — | — | VPP | — |
| VPP | 011 | Bit-line Stress | — | — | VPP | "0" |
| VPP | 100 | OTP Row (after pkg) | $V_{IH}$ | $V_{IH}$ | VPP | data in |
| VPP | 101 | OTP Column (after pkg) | $V_{IH}$ | $V_{IH}$ | VPP | data in |

*The security byte is at $0000 address.

**READ MODE**

The NT68P81's OTP ROM mega cell has 2 control pins. The CE (chip enable) controls the operation power and is used for device selection. The $\overline{OE}$ (output enable) controls the output buffers.

**OUTPUT DISABLE MODE**

If $\overline{OE}$ = $V_{IH}$, the outputs will be in a high impedance state. So two or more ROMs can be connected together on a common bus.

**STANDBY MODE**

By applying a low level to the chip is in standby mode, it will reduce the operating current to 100μA.

**PROGRAM MODE**

Initially, all bits are in "1" state which is an erased state. Thus the program operation is to introduce "0" data into the desired bit locations by electronic programming. When the VPP input is at 12.75V and CE is at $V_{IH}$, the chip is in the PROGRAMMING mode.

**PROGRAM VERLFY MODE**

The VERIFY mode will check to see that the desired data is correctly programmed on the programmed bit. The VERIFY is accomplished with CE at $V_{IH}$, VPP input is at 12.75V, and $\overline{OE}$ = $V_{IL}$.

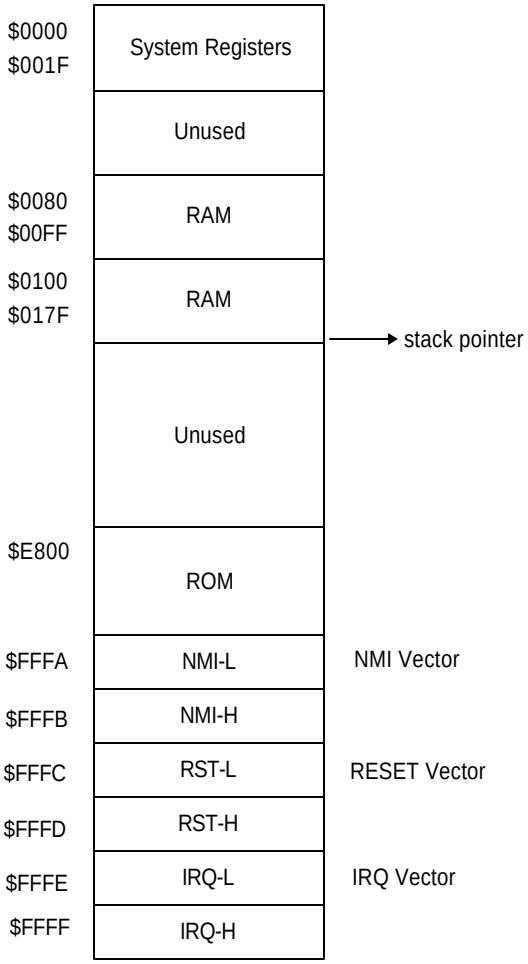
**PROGRAM INHIBIT**

Using this mode, programming of two or more OTP ROMs in parallel with different data is accomplished. All inputs except for CE and $\overline{OE}$ may be commonly connected. The TTL high level program pulse is only applied to the CE of the desired device and TTL high level signal is applied to the other devices.

**4. SRAM: 256 X 8 bits**

The built-in SRAM is used for general purpose data memory and for stack area. SRAM is addressed from 0080H to 017FH. Because the 6502C default stack pointer is 01FFH, the stack area will map $01FF-$0180 to $00FF-$0080, thus the programmer can set the "S" register to 7FH when starting program, allowing stack point to be 017FH.

```
as;   LDX    #$7F
      TXS
```



### 5. System Reserved Registers

| Address | Register | Reset | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $0000 | IRQFUNC | 00H | — | — | — | — | KBD | INT1 | INT0 | TMR | R |
| $0001 | IRQCLRF | 00H | — | — | — | — | CKBD | CINT1 | CINT0 | CTMR | W |
| $0002 | IE_FUNC | 00H | — | — | — | — | EKBD | EINT1 | EINT0 | ETMR | R/W |
| $0003 | IRQUSB | 00H | SUSP | STUP | — | — | IN2 | IN1 | OT0 | IN0 | R |
| $0004 | IRQCLRU | 00H | CSUSP | CSTUP | — | — | CIN2 | CIN1 | COT0 | CIN0 | W |
| $0005 | IE_USB | 00H | ESUSP | ESTUP | — | — | EIN2 | EIN1 | EOT0 | EIN0 | R/W |
| $0006 | BT | 00H | BT7 | BT6 | BT5 | BT4 | BT3 | BT2 | BT1 | BT0 | W |
| $0007 | TCON | 01H | — | — | — | — | — | — | — | $\overline{\text{ENBT}}$ | W |
| $0008 | TMOD | 00H | — | — | — | — | — | TM2 | TM1 | TM0 | R/W |
| $0009 | PORT0 | FFH | P07 | P06 | P05 | P04 | P03 | P02 | P01 | P00 | R/W |
| $000A | PORT1 | FFH | P17 | P16 | P15 | P14 | P13 | P12 | P11 | P10 | R/W |
| $000B | PORT2 | FFH | P27 | P26 | P25 | P24 | P23 | P22 | P21 | P20 | R/W |
| $000C | PORT3 | 1FH | — | — | — | P34 | P33 | P32 | P31 | P30 | R/W |
| $000D | LED | 07H | — | — | — | — | — | LED2 | LED1 | LED0 | W |
| $000E | CLRWDT | 00H | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | W |
| $000F | MODE_FG | 02H | — | — | — | — | — | — | POF | SUSF | R/W |

— : no effect

### 6. Power-on Reset

Built-in power-on reset circuit can generate a minimum of 5ms pulse to reset the entire chip. The user also can use an external $\overline{\text{RESET}}$ pin to reset the entire chip.

### 7. Timing Generator

This block generates the system timing and control signals supplied to the CPU and on-chip peripherals. The crystal oscillator generates a 6MHz system clock. It only generates 3MHz clock for CPU.

## 8. Base Timer (BT)

The Base Timer is an 8-bit counter with a programmable clock source selection. The BT can be enabled/disabled by the CPU. After reset, the BT is disabled and cleared. The BT can be preset by writing a preset value to BT7 ~ BT0 of the BT register at any time. When the BT is enabled, the BT starts counting from the preset value. When the value reaches FFH, it generates a timer interrupt if the timer interrupt is enabled. When it reaches the maximum value of FFH, the BT will wrap around and begin counting at 00H. The BT can be enabled by writing a "0" to "$\overline{\text{ENBT}}$" bit in the TCON (Timer Control) register. The $\overline{\text{ENBT}}$ signal is level trigger.

The input clock source of BT is controlled by the TMOD register. The following table shows 8 ranges of the BT.

| TM2 | TM1 | TM0 | Pre-scalar Ratio | Min. Count | Max. Count |
|---|---|---|---|---|---|
| 0 | 0 | 0 | System Clock/$2^3$ | 1.33 µs | 341.33 µs |
| 0 | 0 | 1 | System Clock/$2^4$ | 2.66 µs | 682.66 µs |
| 0 | 1 | 0 | System Clock/$2^5$ | 5.32 µs | 1.36 ms |
| 0 | 1 | 1 | System Clock/$2^6$ | 10.64 µs | 2.72 ms |
| 1 | 0 | 0 | System Clock/$2^7$ | 21.28 µs | 5.44 ms |
| 1 | 0 | 1 | System Clock/$2^8$ | 42.56 µs | 10.89 ms |
| 1 | 1 | 0 | System Clock/$2^9$ | 85.12 µs | 21.79 ms |
| 1 | 1 | 1 | System Clock/$2^{10}$ | 170.24 µs | 43.58 ms |

For counting accuracy, please set the TMOD register first, then preset the BT register, and enable the base timer finally.
**(TM2, TM1, TM0) = (1, 1, 1) is reserved for USB driver use.**

## 9. Interrupt Controller

There are 10 interrupt sources: Timer, INT0, INT1, KBD, SUSP, IN0, IN1, IN2, OT0 and STUP.

### 9.1. Timer Interrupt

When the BASE TIMER overflows, it will set the TMR flag, If the interrupt is enabled by writing "1" to the bit 0 in IE_FUNC ($0002H), then it will interrupt 6502 CPU. The TMR flag can be read by the software. Once set by an interrupt source, it can read from bit0 in IRQFUNC ($0000H) and remains high unless cleared by writing "1" to the bit 0 in IRQCLRF ($0001H). All of register's data is cleared to "0" at initialization by the system reset. When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, thus the TMR flag must be cleared by the software.

### 9.2. INT0 Interrupt

As soon as INT0 pin detects a falling edge trigger, NT68P81 sets the INT0 flag ($0000H, bit1). After that, the 6502 CPU is interrupted if this interrupt has been already been enabled by writing "1" to EINT0 ($0002H, bit1). If the EINT0 flag is cleared, the 6502 CPU can' t be INT0 interrupted even if the INT0 flag is set. INT0 flag can be only be set by hardware and cannot be set or cleared directly by the software except for writing "1" to CINT0 ($0001H, bit1) flag to clear INT0 flag. When an interrupt occurs, the CPU will jump to $FFFEH & $FFFFH to execute the interrupt service routine so the INT0 flag must be cleared by software.

### 9.3. INT1 Interrupt

As soon as the INT1 pin detects a falling edge trigger, NT68P81 sets the INT1 flag ($0000H, bit2). Then the 6502 CPU is interrupted if the interrupt has already been enabled by writing "1" to EINT1 ($0002H, bit2). If EINT0 flag is cleared, the 6502 CPU can' t be INT1 interrupted even if INT1 flag is set. INT1 flag can only be set by the hardware and can not be set or cleared directly by the software except for writing "1" to CINT1 ($0001H, bit2) flag to clear INT1 flag.

When an interrupt occurs, CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the INT1 flag must be cleared by the software.

### 9.4. KBD Interrupt

This interrupt will set the KBD flag ($0000H, bit3) every 4ms(HID 1.00 version) to indicate that keyboard scan data is ready to send for endpoint1. Then the 6502 CPU is interrupted if this interrupt has been enabled already by writing "1" to EKBD ($0002H, bit3). If the EKBD flag is cleared, the 6502 CPU can' t be KBD interrupted even if the KBD flag is set. The KBD flag can only be set by the hardware and can not be set or cleared directly by the software except for writing "1" to CKBD ($0001H, bit 3) flag to clear KBD flag.

When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the KBD flag must be cleared by the software.

### 9.5. IN0 Token Interrupt

When an IN TOKEN for endpoint 0 is done, it will set the IN0 flag. If this interrupt is enabled by writing "1" to EIN0 ($0005H, bit0), it will interrupt 6502 CPU.

When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the IN0 flag must be cleared by the software.

### 9.6. OT0 (OUT 0) Token Interrupt

When an OUT TOKEN for endpoint 0 is done, it will set the OT0 flag. If this interrupt is enabled by writing "1" to EOT0 ($0005H, bit1), it will interrupt 6502 CPU.

When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the OT0 flag must be cleared by the software.

### 9.7. IN1 Token Interrupt

When an IN TOKEN for endpoint 1 is done, it will set the IN1 flag. If this interrupt is enabled by writing "1" to EIN1 ($0005H, bit2), it will interrupt the 6502 CPU.

When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the IN1 flag must be cleared by the software.

### 9.8. IN2 Token Interrupt

When an IN TOKEN for endpoint 2 is done, it will set the IN2 flag. If this interrupt is enabled by writing "1" to EIN2 ($0005H, bit3), it will interrupt 6502 CPU.

When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the IN2 flag must be cleared by the software.

### 9.9. STUP (SETUP) Token Interrupt

When a SETUP TOKEN for endpoint 0 is done, it will set the STUP flag. If this interrupt is enabled by writing "1" to ESTUP ($0005H, bit6), it will interrupt 6502 CPU.

When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the STUP flag must be cleared by the software.

### 9.10. SUSP Interrupt

When USB SIE detects a suspend signal, it sets the SUSP flag. Then the 6502 CPU is interrupted if the interrupt has already been enabled by writing "1" to ESUSP ($0005H, bit7). If ESUSP flag is cleared, 6502 CPU can't be SUSP interrupted even if SUSP flag is set. SUSP flag can be set by H/W only and can't be set/cleared directly by the software except for writing "1" to CSUSP ($0004H, bit 7) flag to clear SUSP flag.

When an interrupt occurs, the CPU jumps to $FFFEH & $FFFFH to execute the interrupt service routine, the SUSP flag must be cleared by the software.

## 10. I/O PORTs

The NT68P81 has 32 pins dedicated to input and output. These pins are grouped into 5 ports, as follows:

**PORT0 (P00~P07)**

PORT0 is an 8-bit bi-directional CMOS I/O port that is internally pulled high by PMOS. Each pin of PORT0 can be bit programmed as an input or output port under software control. When programmed as output, data is latched to the port data register and output to the pin. PORT0 pins with "1" written to them are pulled high by the internal PMOS pull-ups, and can be used as inputs in that state, then these input signals can be read. The port will output high after the reset.

**PORT1 (P10~P17)**: Functions the same as PORT0.
**PORT2 (P20~P27)**: Functions the same as PORT0.
**PORT3 (P30~P34)**: Functions the same as PORT0. Except for P33/P32 is shared with INT1/INT0 pin. It is also a Schmitt Trigger input with an interrupt source of falling edge sensitive.

**LED:** There are three LED direct sink pins which require no external serial resistors.
The address is mapped to $000DH.

## 11. Watch-Dog Timer (WDT)

The NT68P81 has a watch-dog timer reset function that protects programs against system standstill. The clock of the WDT is derived from the crystal oscillator. The WDT interval is about 0.15 seconds when the operation frequency is 6MHz. The timer must be cleared every 0.15 second during normal operation; otherwise, it will overflow and cause a system reset (This cannot be disabled by the software). Before watch-dog reset occurs, the software will clear the watch-dog register by writing #55H to CLRWDT ($000EH) register.

For example:

```
LDA    #$55H
STA    $000E
```

## 12. Power Control

The power-off flag (POF) in the MODE_FG register indicates whether a reset is a warm start or a cold start reset. POF is set by hardware when an external power $V_{cc}$ arises to its normal operating level, and must be cleared by the software in the cold reset initialization procedure. A warm start reset (POF = 0) occurs at a watch-dog reset or resume reset.

| Address | Register | Reset | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $000FH | MODE_FG | 02H | - | - | - | - | - | - | POF | SUSF | R/W |

## 13. Universal Serial Bus Interface

Please refer to the UNIVERSAL SERIAL BUS specification Version 1.0 Chapter 7, 8, and 9.

## 14. Suspend and Resume

**Suspend:**

When SIE receives the suspend signal, NT68P81 generates a SUSP interrupt request. In the SUSP interrupt service routine, the software will carry out the following steps:

1) Clear SUSP IRQ flag,
2) Store all the port status,
3) Force return lines (PORT2) pull-high,
4) Force scan lines (PORT0, PORT1 and P30, P31 or P32) pull-low,
5) Turn off LED output,
6) Clear watch-dog register

After the above action has been completed, the software will then set SUSLO ($1EH) to #55H and SUSHI ($1FH) to #AAH in order to enter the SUSPEND mode. The oscillator will stop for in order to save power.

**Resume:**

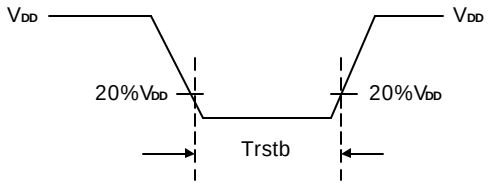
When the SIE detects a resume signal, the NT68P81 trigger oscillator to oscillate and resets whole chip. After a reset, software checks the status of POF bit in MODE_FG register to see whether a cold start reset or a warm start reset occurred. If cold reset, it executes all initial procedure. If warm reset, software checks the status of SUSF bit in MODE_FG register to see whether a watch-dog reset or resume reset. Under resume reset condition, programmer should restores all port status. After a warm start, user software should clear the SUSF bit. When any key stroked in suspend mode, it remotely resume NT68P81 functions. The action is same as host resume.

## 15. Reset Source Summary

These are 5 reset sources in NT68P81 as shown below.

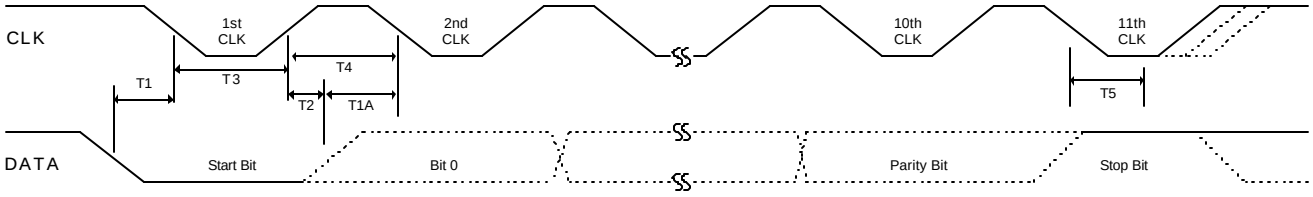
| No. | Type | Function | Description |
|---|---|---|---|
| 1 | Cold | External Pin ( $\overline{\text{RESET}}$ ) | Applied Externally |
| 2 | Cold | Power-on Reset | Reset after Power-on |
| 3 | Cold | USB Reset Signaling | 10 ms Reset Period |
| 4 | Warm-1 | Resume Reset | USB Reset Period |
| 5 | Warm-2 | Watch-dog Reset | Reset every 0.15S (OSC = 6MHz) |

NT68P81 can also be reset externally through the $\overline{\text{RESET}}$ pin. A reset is initialed when the signal at the $\overline{\text{RESET}}$ pin is held Low for at least 10 system clocks. When $\overline{\text{RESET}}$ signal goes high, the NT68P81 begins to work. The following shows the definition of $\overline{\text{RESET}}$ input low pulse width.



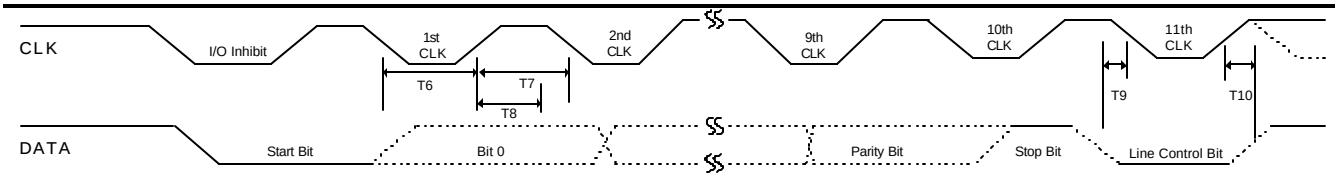

## 16. PS/2 Mouse Application

A PS/2 mouse interface is implemented in P32 (CLK), P33 (DATA) and P34 (Power Control). The timing diagrams are described as follows.




Auxiliary Device Sending Data Timings

| Timing | Description | MIN/MAX |
|---|---|---|
| T1 | Time from DATA transaction to falling edge of CLK 1 | 5/25*u* s |
| T1A | Time from DATA transaction to falling edge of CLK 2-11 | 5/25*u* s |
| T2 | Time from rising edge of CLK to DATA transaction | 5/T4-5*u* s |
| T3 | Duration of CLK inactive (LOW) | 30/50*u* s |
| T4 | Duration of CLK active (HIGH) | 30-50*u* s |
| T5 | Time to Auxiliary Device inhibit after clock 11 to ensure the Auxiliary Device does not start another transmission | >0/50*u* s |




Auxiliary Device Receiving Data Timings

| Timing | Description | MIN/MAX |
|---|---|---|
| T6 | Duration of CLK interface (LOW) | 30/50*u* s |
| T7 | Duration of CLK active (HIGH) | 30/50*u* s |
| T8 | Time from inactive to active CLK transition, used to time when the Auxiliary Device samples DATA | 5/25*u* s |
| T9 | Time from falling edge of line control bit to falling edge of clock 11 CLK | 5*u* s/ |
| T10 | Time from rising edge of clock 11 to rising edge of line control bit | 5/25*u* s |

### Absolute Maximum Rating*

DC Supply Voltage . . . . . -0.3V to +7.0V

Input/Output Voltage . . . . . .GND - 0.2V to $V_{DD}$ + 0.2V

Operating Ambient Temperature . . . . . .0 °C to 70 °C

Storage Temperature . . . . . .-55 °C to +125 °C

Operating Voltage ($V_{DD}$) . . . . . .+4.4V to +5.25V

### *Comments

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied and exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Electrical characteristics ($V_{DD}$ = 5V, GND = 0V, $T_A$ = 25°C, $F_{OSC}$ = 6MHz, unless otherwise noted)

| Symbol | Parameters | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|
| $V_{DD}$ | Operating Voltage | 4.4 | 5 | 5.25 | V | |
| $I_{OP}$ | Operating Current | | | 20 | mA | No load |
| $I_{SP}$ | Suspend Current | | | 500 | µA | |
| $V_{IH}$ | Input High Voltage | 2 | | | V | |
| $V_{IL}$ | Input Low Voltage | | | 0.8 | V | |
| $V_{OH}$ | Output High Voltage | 2.4 | | | V | $I_{OH}$ = -100µA |
| $V_{OL1}$ | Output Low Voltage (P0/P1/P2) | | | 0.4 | V | $I_{OL1}$ = 4mA |
| $V_{OL2}$ | Output Low Voltage (P3) | | | 0.4 | V | $I_{OL2}$ = 5mA |
| $I_{LED}$ | LED Sink Current | 6 | 10 | 14 | mA | $V_{OL}$ = 3.2V |
| $V_{STIH}$ | Schmitt Trigger Input High Voltage | | 1.7 | 2 | V | |
| $V_{STIH}$ | Schmitt Trigger Input Low Voltage | 0.8 | 1.1 | | V | |

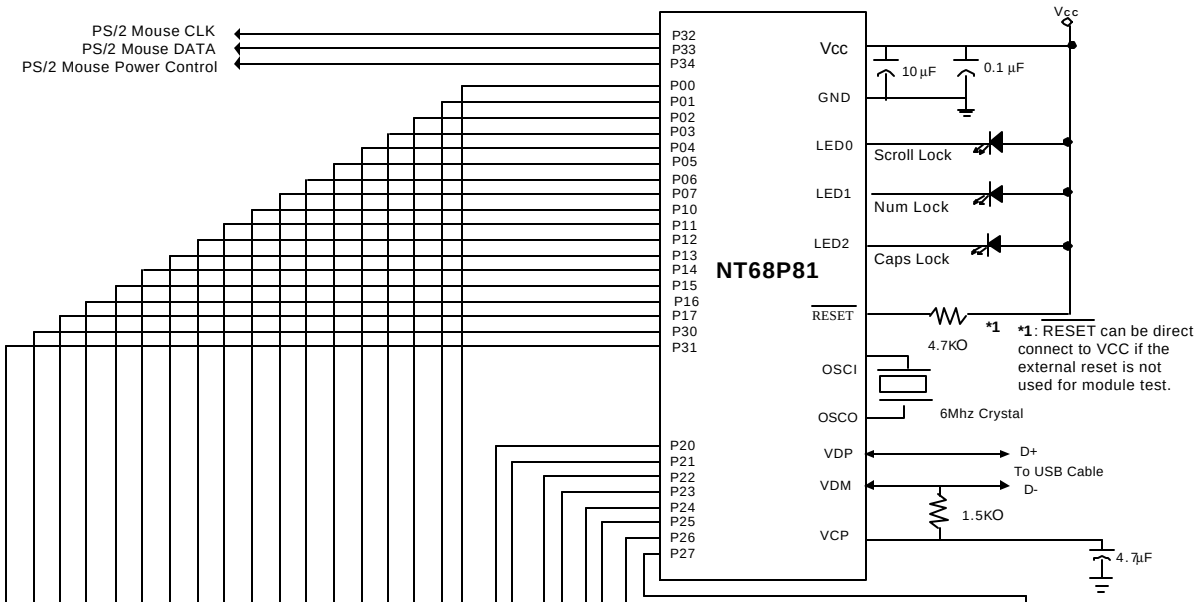
### AC Electrical characteristics ($V_{DD}$ = 5V, GND = 0V, $T_A$ = 25°C, $F_{OSC}$ = 6MHz, unless otherwise noted)

| Symbol | Parameters | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|
| $F_{OSC}$ | Oscillator Frequency | 5.97 | 6 | 6.03 | MHz | OSC within +/- 0.5% |
| $T_{RSTB}$ | $\overline{\text{RESET}}$ Input Low Pulse Width | 1.67 | | | µs | 10 system clocks |
| $T_{POR}$ | Power-on Reset Time | 5 | | 30 | ms | |

### USB DC/AC SPECIFICATIONS

Please refer to the UNIVERSAL SERIAL BUS specification Version 1.0 Chapter 7.

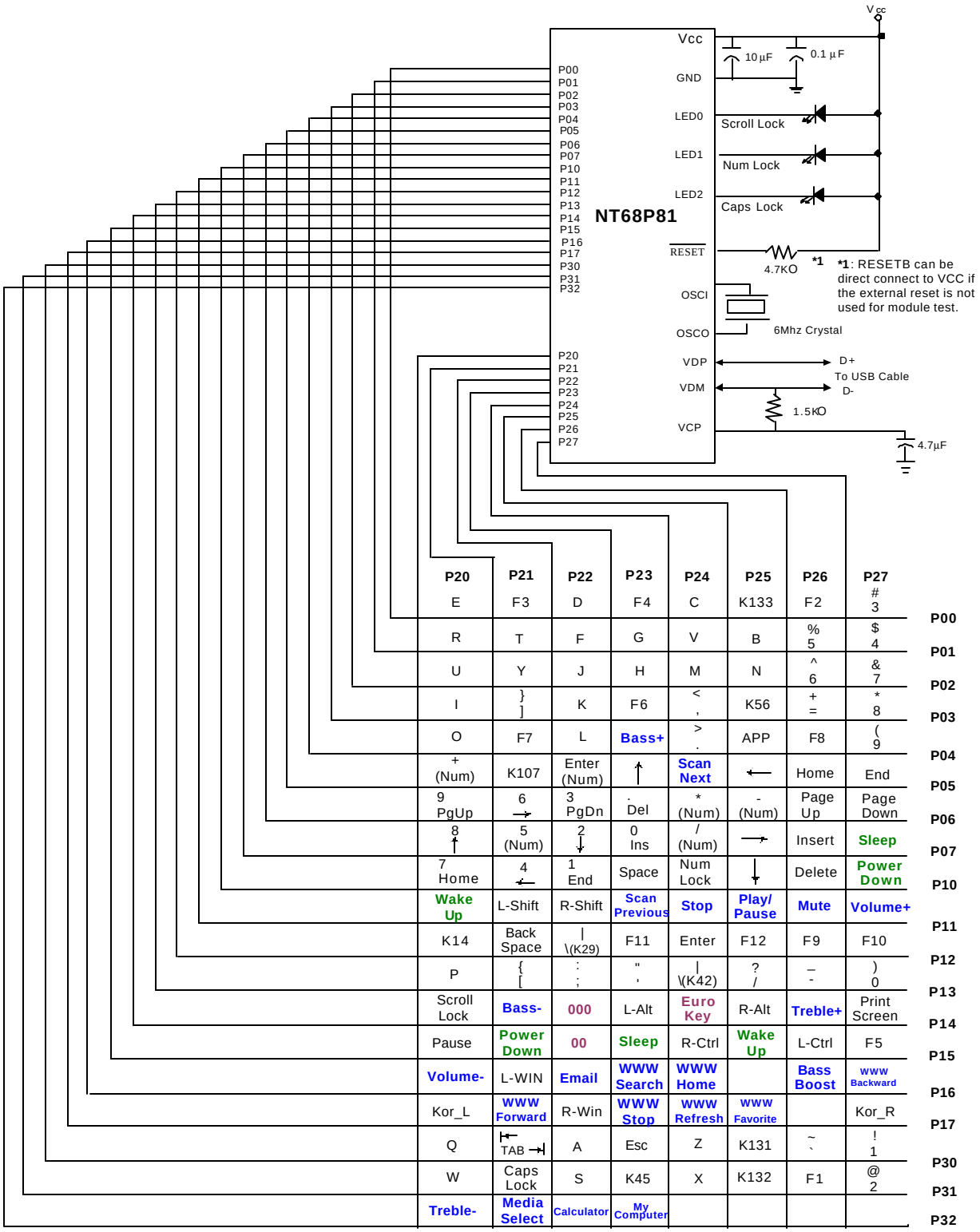
## Application Circuit 1 (Simple Keyboard with PS/2 Mouse)




| P20 | P21 | P22 | P23 | P24 | P25 | P26 | P27 | |
|---|---|---|---|---|---|---|---|---|
| E | F3 | D | F4 | C | K133 | F2 | # 3 | |
| R | T | F | G | V | B | % 5 | $ 4 | P00 |
| U | Y | J | H | M | N | ^ 6 | & 7 | P01 |
| I | } ] | K | F6 | < , | K56 | + = | * 8 | P02 |
| O | F7 | L | | > . | APP | F8 | ( 9 | P03 |
| + (Num) | K107 | Enter (Num) | ↑ | | ← | Home | End | P04 |
| 9 PgUp | 6 → | 3 PgDn | . Del | * (Num) | - (Num) | Page Up | Page Down | P05 |
| 8 ↑ | 5 (Num) | 2 ↓ | 0 Ins | / (Num) | → | Insert | Kor_R | P06 |
| 7 Home | 4 ← | 1 End | Space | Num Lock | ↓ | Delete | Kor_L | P07 |
| | L-Shift | R-Shift | | | | | | P10 |
| K14 | Back Space | \| \(K29) | F11 | Enter | F12 | F9 | F10 | P11 |
| P | { [ | : ; | " ' | \| \(K42) | ? / | _ - | ) 0 | P12 |
| Scroll Lock | | 000 | L-Alt | | R-Alt | | Print Screen | P13 |
| Pause | | 00 | | R-Ctrl | | L-Ctrl | F5 | P14 |
| | L-WIN | | | | | | | P15 |
| Kor_L | | R-Win | | | | | Kor_R | P16 |
| Q | TAB → \| | A | Esc | Z | K131 | ~ ` | ! 1 | P17 |
| W | Caps Lock | S | K45 | X | K132 | F1 | @ 2 | P30 |
| | | | | | | | | P31 |

Notice: "Return Key" must be forced to PORT2 for remote wake up function. If not, remote wake up function will not work.

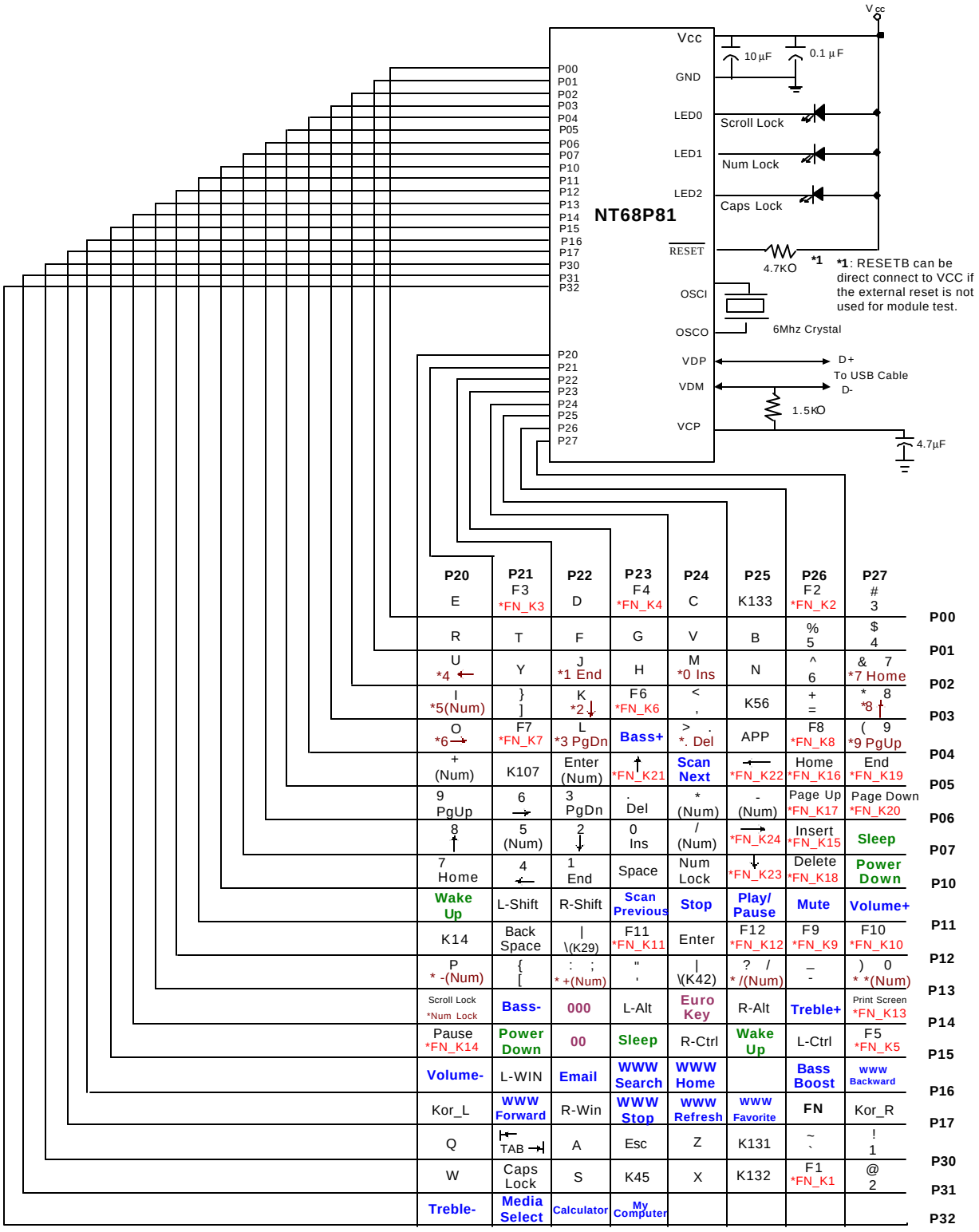
## Application Circuit 2 (Windows 2000 Compatible Keyboard)




| | P20 | P21 | P22 | P23 | P24 | P25 | P26 | P27 | |
|---|---|---|---|---|---|---|---|---|---|
| | E | F3 | D | F4 | C | K133 | F2 | # 3 | |
| P00 | R | T | F | G | V | B | % 5 | $ 4 | P00 |
| P01 | U | Y | J | H | M | N | ^ 6 | & 7 | P01 |
| P02 | I | } ] | K | F6 | < , | K56 | + = | * 8 | P02 |
| P03 | O | F7 | L | Bass+ | > . | APP | F8 | ( 9 | P03 |
| P04 | + (Num) | K107 | Enter (Num) | ↑ | Scan Next | ← | Home | End | P04 |
| P05 | 9 PgUp | 6 → | 3 PgDn | . Del | * (Num) | - (Num) | Page Up | Page Down | P05 |
| P06 | 8 ↑ | 5 (Num) | 2 ↓ | 0 Ins | / (Num) | → | Insert | Sleep | P06 |
| P07 | 7 Home | 4 ← | 1 End | Space | Num Lock | ↓ | Delete | Power Down | P07 |
| P10 | Wake Up | L-Shift | R-Shift | Scan Previous | Stop | Play/ Pause | Mute | Volume+ | P10 |
| P11 | K14 | Back Space | \| \(K29) | F11 | Enter | F12 | F9 | F10 | P11 |
| P12 | P | { [ | : ; | " ' | \| \(K42) | ? / | _ - | ) 0 | P12 |
| P13 | Scroll Lock | Bass- | 000 | L-Alt | Euro Key | R-Alt | Treble+ | Print Screen | P13 |
| P14 | Pause | Power Down | 00 | Sleep | R-Ctrl | Wake Up | L-Ctrl | F5 | P14 |
| P15 | Volume- | L-WIN | Email | WWW Search | WWW Home | | Bass Boost | WWW Backward | P15 |
| P16 | Kor_L | WWW Forward | R-Win | WWW Stop | WWW Refresh | WWW Favorite | | Kor_R | P16 |
| P17 | Q | ↹ TAB | A | Esc | Z | K131 | ~ ` | ! 1 | P17 |
| P30 | W | Caps Lock | S | K45 | X | K132 | F1 | @ 2 | P30 |
| P31 | Treble- | Media Select | Calculator | My Computer | | | | | P31 |
| P32 | | | | | | | | | P32 |

Notice: “Return Key” must be forced to PORT2 for remote wake up function. If not, remote wake up function will not work.

## Application Circuit 3 (Mini Keyboard)




| P20 | P21 | P22 | P23 | P24 | P25 | P26 | P27 | |
|---|---|---|---|---|---|---|---|---|
| E | F3 *FN_K3 | D | F4 *FN_K4 | C | K133 | F2 *FN_K2 | # 3 | P00 |
| R | T | F | G | V | B | % 5 | $ 4 | P01 |
| U *4 ← | Y | J *1 End | H | M *0 Ins | N | ^ 6 | & 7 *7 Home | P02 |
| I *5(Num) | } ] | K *2 ↓ | F6 *FN_K6 | < , | K56 | + = | * 8 *8 ↑ | P03 |
| O *6 → | F7 *FN_K7 | L *3 PgDn | Bass+ | > . *. Del | APP | F8 *FN_K8 | ( 9 *9 PgUp | P04 |
| + (Num) | K107 | Enter (Num) | ↑ *FN_K21 | Scan Next | ← *FN_K22 | Home *FN_K16 | End *FN_K19 | P05 |
| 9 PgUp | 6 → | 3 PgDn | . Del | * (Num) | - (Num) | Page Up *FN_K17 | Page Down *FN_K20 | P06 |
| 8 ↑ | 5 (Num) | 2 ↓ | 0 Ins | / (Num) | → *FN_K24 | Insert *FN_K15 | Sleep | P07 |
| 7 Home | 4 ← | 1 End | Space | Num Lock | ↓ *FN_K23 | Delete *FN_K18 | Power Down | P10 |
| Wake Up | L-Shift | R-Shift | Scan Previous | Stop | Play/ Pause | Mute | Volume+ | P11 |
| K14 | Back Space | \| \(K29) | F11 *FN_K11 | Enter | F12 *FN_K12 | F9 *FN_K9 | F10 *FN_K10 | P12 |
| P * -(Num) | { [ | : ; * +(Num) | " ' | \| \(K42) | ? / * /(Num) | _ - | ) 0 * *(Num) | P13 |
| Scroll Lock *Num Lock | Bass- | 000 | L-Alt | Euro Key | R-Alt | Treble+ | Print Screen *FN_K13 | P14 |
| Pause *FN_K14 | Power Down | 00 | Sleep | R-Ctrl | Wake Up | L-Ctrl | F5 *FN_K5 | P15 |
| Volume- | L-WIN | Email | WWW Search | WWW Home | | Bass Boost | www Backward | P16 |
| Kor_L | WWW Forward | R-Win | WWW Stop | WWW Refresh | www Favorite | FN | Kor_R | P17 |
| Q | TAB → | A | Esc | Z | K131 | ~ ` | ! 1 | P30 |
| W | Caps Lock | S | K45 | X | K132 | F1 *FN_K1 | @ 2 | P31 |
| Treble- | Media Select | Calculator | My Computer | | | | | P32 |

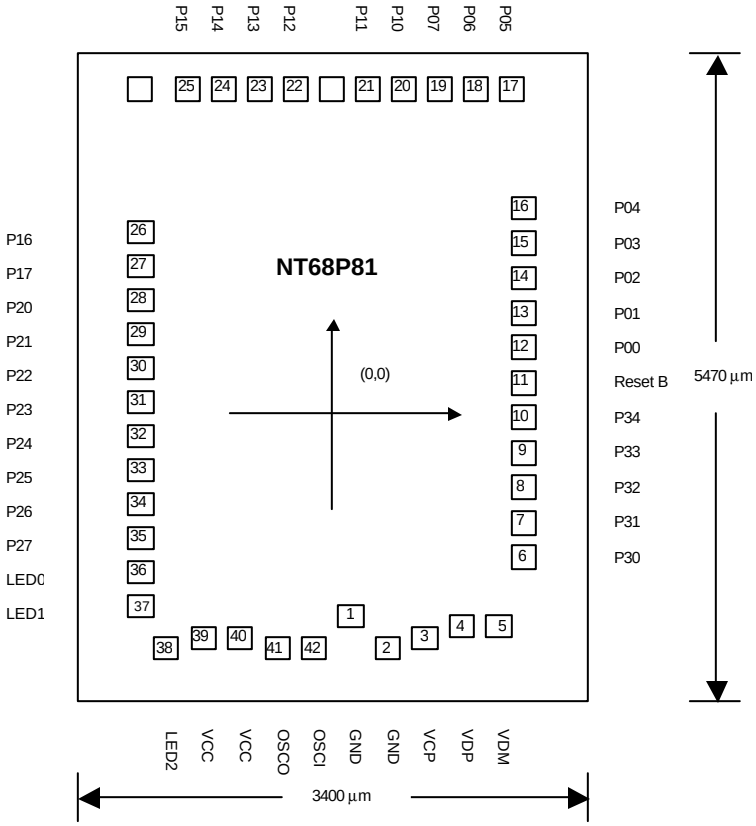
Notice: “Return Key” must be forced to PORT2 for remote wake up function. If not, remote wake up function will not work.

*: For FN key model usage

| FN Key Model Usage for Keypad | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FN+Scroll Lock | Num Lock | | | | | | | | | | |
| FN+& | 7 | 7 Home | FN+* | 8 | 8 ↑ | FN + ( | 9 | 9 PgUp | FN+) | 0 | *(Num) |
| FN+U | | 4 ← | FN+I | | 5(Num) | FN+O | | 6 → | FN+P | | -(Num) |
| FN+J | | 1 End | FN+K | | 2 ↓ | FN+L | | 3 PgDn | FN+: | ; | +(Num) |
| FN+M | | 0 Ins | | | | FN+> | . | . Del | FN+? | / | /(Num) |

| FN Key Model Usage for Consumer Keys | | | | | |
|---|---|---|---|---|---|
| FN_K1 | FN+F1 | WWW Backward | FN_K2 | FN+F2 | WWW Forward |
| FN_K3 | FN+F3 | WWW Stop | FN_K4 | FN+F4 | WWW Refresh |
| FN_K5 | FN+F5 | WWW Search | FN_K6 | FN+F6 | WWW Favorite |
| FN_K7 | FN+F7 | WWW Home | FN_K8 | FN+F8 | Email |
| FN_K9 | FN+F9 | My Computer | FN_K10 | FN+F10 | Calculator |
| FN_K11 | FN+F11 | Media Select | FN_K12 | FN+F12 | Mute |
| FN_K13 | FN+Print Screen | Bass Boost | FN_K14 | FN+Pause | Sleep |
| FN_K15 | FN+Insert | Volume+ | FN_K16 | FN+Home | Bass+ |
| FN_K17 | FN+Page Up | Treble+ | FN_K18 | FN+Delete | Volume- |
| FN_K19 | FN+End | Bass- | FN_K20 | FN+Page Down | Treble- |
| FN_K21 | FN+ ↑ | Stop | FN_K22 | FN+ ← | Scan Previous Track |
| FN_K23 | FN+ ↓ | Play/Pause | FN_K24 | FN+ → | Scan Next Track |

## Bonding Diagram




Substrate connect to VCC

Unit: µm

| Pad No. | Designation | X | Y |
|---|---|---|---|
| 1 | GND | 264.50 | -2460.05 |
| 2 | GND | 424.50 | -2481.00 |
| 3 | VCP | 734.95 | -2470.00 |
| 4 | VDP | 1069.35 | -2466.00 |
| 5 | VDM | 1368.85 | -2466.00 |
| 6 | P30 | 1443.05 | -2069.05 |
| 7 | P31 | 1443.05 | -1768.65 |
| 8 | P32 | 1443.05 | -1468.25 |
| 9 | P33 | 1443.05 | -1167.85 |
| 10 | P34 | 1443.05 | -867.45 |
| 11 | $\overline{\text{RESET}}$ | 1443.05 | -560.45 |
| 12 | P00 | 1443.05 | -235.35 |
| 13 | P01 | 1443.05 | 65.05 |
| 14 | P02 | 1443.05 | 365.45 |
| 15 | P03 | 1443.05 | 659.85 |
| 16 | P04 | 1443.05 | 1007.20 |
| 17 | P05 | 1463.90 | 2545.00 |
| 18 | P06 | 1173.90 | 2545.00 |
| 19 | P07 | 883.90 | 2545.00 |
| 20 | P10 | 593.90 | 2545.00 |
| 21 | P11 | 303.90 | 2545.00 |
| **Pad No.** | **Designation** | **X** | **Y** |
| 22 | P12 | -308.10 | 2545.00 |
| 23 | P13 | -598.10 | 2545.00 |
| 24 | P14 | -888.10 | 2545.00 |
| 25 | P15 | -1178.10 | 2545.00 |
| 26 | P16 | -1478.00 | 1012.50 |
| 27 | P17 | -1478.00 | 670.80 |
| 28 | P20 | -1478.00 | 370.40 |
| 29 | P21 | -1478.00 | 70.00 |
| 30 | P22 | -1478.00 | -230.40 |
| 31 | P23 | -1478.00 | -530.80 |
| 32 | P24 | -1478.00 | -831.20 |
| 33 | P25 | -1478.00 | -1131.60 |
| 34 | P26 | -1478.00 | -1432.00 |
| 35 | P27 | -1478.00 | -1732.40 |
| 36 | LED0 | -1478.00 | -2037.15 |
| 37 | LED1 | -1478.00 | -2337.55 |
| 38 | LED2 | -1112.85 | -2481.00 |
| 39 | VCC | -812.45 | -2470.55 |
| 40 | VCC | -626.85 | -2470.55 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 41 | OSCO | -326.45 | -2481.00 | 42 | OSCI | -7.75 | -2481.00 |

## Ordering Information

| Part No. | Packages |
|---|---|
| NT68P81H | CHIP FORM |
| NT68P81 | 40L DIP |

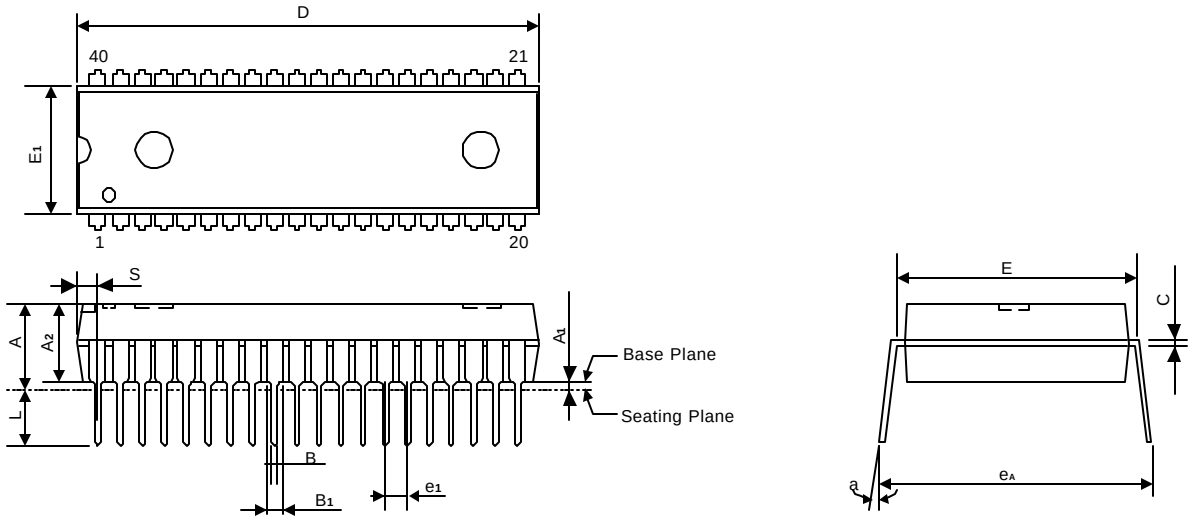
## Standard code functional descriptions

| Code Number | Name | Reference application circuit | Functional Description |
|---|---|---|---|
| NT68P81-D01012 | Simple Keyboard with PS/2 Mouse | Application circuit 1 | 1. PS/2 mouse port<br>2. '000' and '00' keys |
| NT68P81-D01013 | Windows 2000 Compatible Keyboard | Application circuit 2 | 1. ACPI keys<br>2. '000', '00' and Euro keys<br>3. Consumer keys (Windows 2000) |
| NT68P81-D01014 | Mini Keyboard | Application circuit 3 | 1. ACPI keys<br>2. ‘000’, ‘00’ and Euro keys<br>3. Consumer keys (Windows 2000)<br>4. FN key and 40 Translated keys |

## Package Information

### P-DIP 40L Outline Dimensions

unit: inches/mm




| Symbol | Dimensions in inches | Dimensions in mm |
|---|---|---|
| A | 0.210 Max. | 5.33 Max. |
| $A_1$ | 0.010 Min. | 0.25 Min. |
| $A_2$ | 0.155±0.010 | 3.94±0.25 |
| B | 0.018 +0.004 -0.002 | 0.46 +0.10 -0.05 |
| $B_1$ | 0.050 +0.004 -0.002 | 1.27 +0.10 -0.05 |
| C | 0.010 +0.004 -0.002 | 0.25 +0.10 -0.05 |
| D | 2.055 Typ. (2.075 Max.) | 52.20 Typ. (52.71 Max.) |
| E | 0.600±0.010 | 15.24±0.25 |
| $E_1$ | 0.550 Typ. (0.562 Max.) | 13.97 Typ. (14.27 Max.) |
| $e_1$ | 0.100±0.010 | 2.54±0.25 |
| L | 0.130±0.010 | 3.30±0.25 |
| α | 0°~ 15° | 0°~ 15° |
| $e_A$ | 0.655±0.035 | 16.64±0.89 |
| S | 0.093 Max. | 2.36 Max. |

**Note:**

1. The maximum value of dimension D includes end flash.
2. Dimension $E_1$ does not include resin fins.
3. Dimension S includes end flash.

**Product Spec. Change Notice**

| NT68P81 Specification Revision History | | |
|---|---|---|
| **Version** | **Content** | **Data** |
| 2.1 | FN Key Model Usage for Consumer Keys modified - FN_K22 and FN_K24 (Page 24) | Oct. 2002 |
| 2.0 | Volume Knob Application deleted (Page 18)<br>PS/2 Mouse Application added (Page 18 and 19)<br>Application circuit 2 and 3 modified (Page 22 and 23)<br>FN key usage added (Page 24)<br>Standard code functional descriptions modified (Page 26) | Sep. 2002 |
| 1.3 | Application circuits modified (Page 20, 21 and 22)<br>Standard code functional description added (Page 24) | July 2002 |
| 1.0 | Original | Nov. 1998 |