8051 Ports and External Memory access

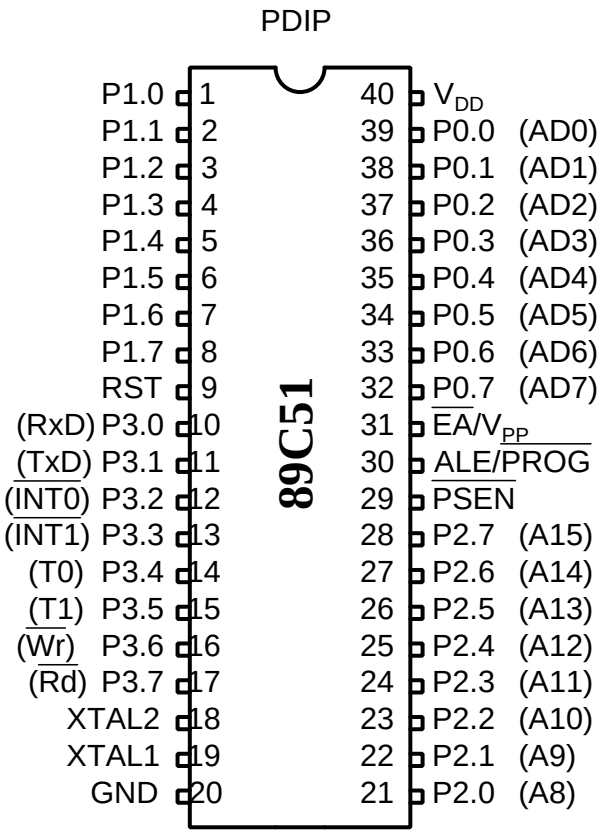
Dinesh Sharma

Department Of Electrical Engineering
Indian Institute Of Technology, Bombay

February 3, 2021

8051 Pin Diagram

The pin diagram of a 40 pin package for an 8051 is shown below:

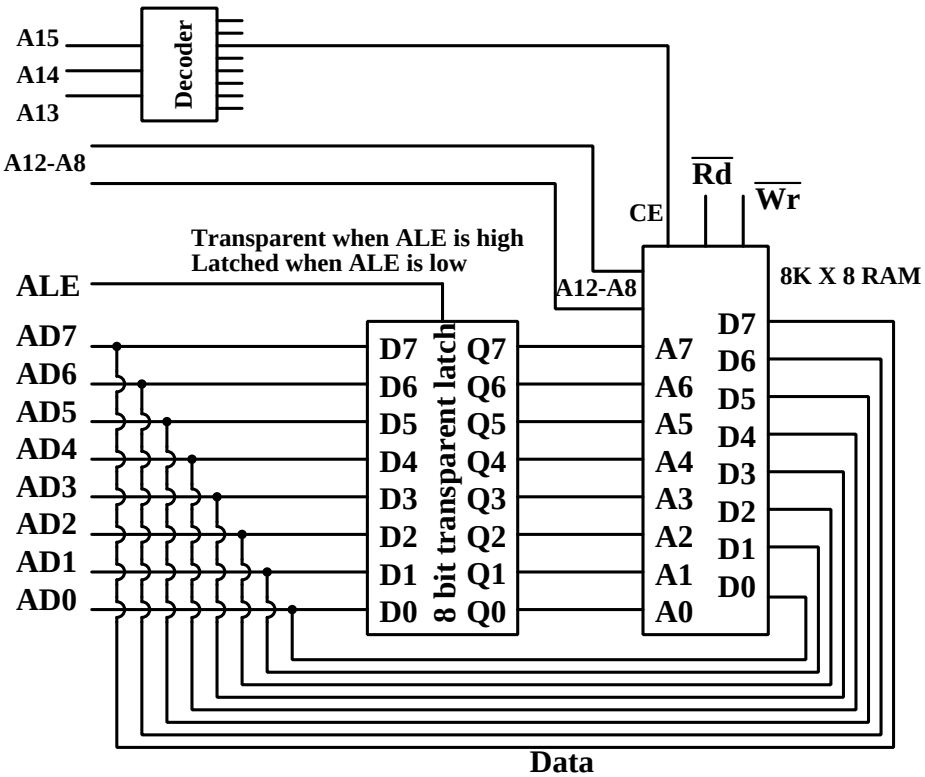


- Only port 1 has dedicated pins (1-8) allotted to it. Other port pins get used as the bus for external memory when pin 31 ($\overline{EA}/V_{PP}$) is grounded.
- Alternative function of port 2 pins (21-28) is to provide the more significant byte of the memory address.
- Lower byte of address and data are multiplexed on Port 0 pins (32-39) using the ALE signal on pin 30.

Pins 16 and 17 (otherwise meant for bits 6 and 7 of port 3) are used as $\overline{Rd}$ and $\overline{Wr}$ control signals for external RAM when ($\overline{EA}/V_{PP}$) is grounded.

Demultiplexing Address and Data

When external memory must be used, ports 0 and 2 are used to provide the external data bus with multiplexed address and data replacing Port 0 and the higher address pins replacing Port 2.



- AD7-AD0 are fed to an 8 bit transparent latch.
- The Latch is transparent when ALE is high, latched when ALE is low.
- Q outputs of the latch have the de-multiplexed address A7-A0

We use $\overline{\text{Psen}}$ instead of $\overline{\text{Rd}}$ for an external ROM.

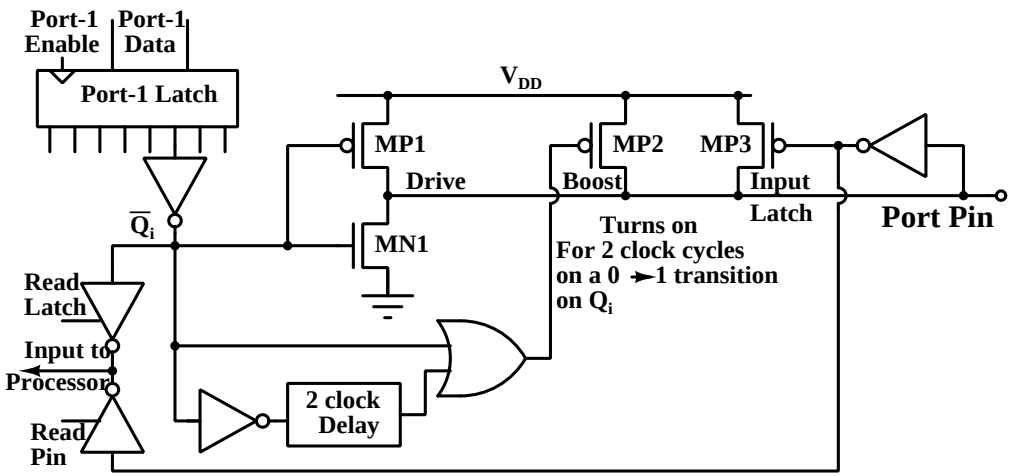
Port-1 driver

Pins 1-8 of 8051 are dedicated to port-1. The port may be used to output data or for reading data from the external world.

- For using the port for output, data is written to the port latch (at address 90H) and it is buffered to the port pins.
- For using the port for input, a '1' is written to every bit. Now, if the external circuit wants to send a '1', there is no problem (We have written a '1' to port-1 latch and its buffer is already driving the pin to HIGH).
- However, if the external circuit wants to send a '0', it needs to fight the '1' output of the latch and pull the pin down to '0' by sinking current.
- This presents a design problem:
  If the output drive is strong, the external circuit may find it hard to pull it down to '0' by sinking current.
  If the output drive is made weak, $0 \rightarrow 1$ transitions will be slow when the port is used as an output port.

Port-1 driver-Boost circuit

The problem is solved by using a clever circuit. The pull up circuit for the pin includes a strong pMOS transistor (MP2) driven by an OR gate.

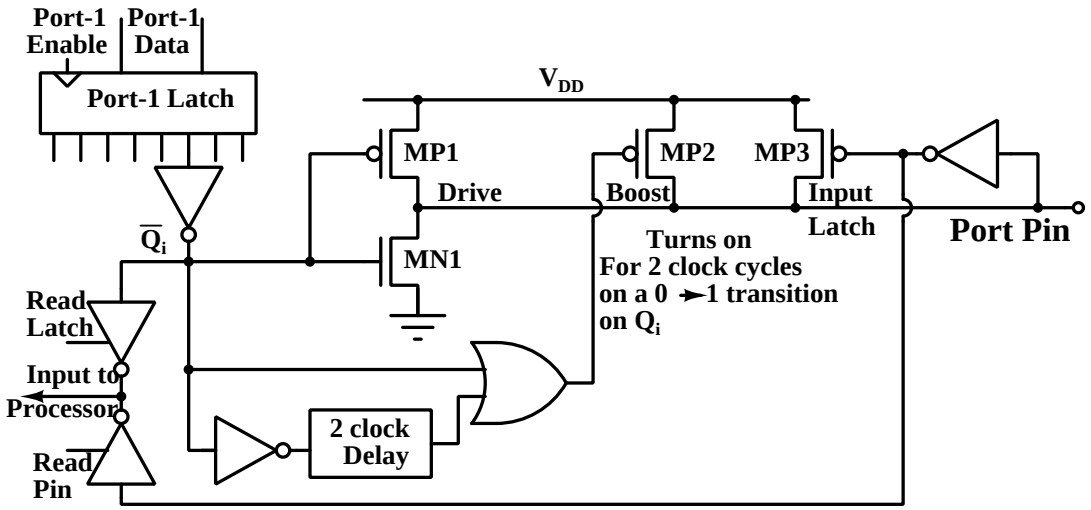


Port-1 Driver (There is one such for every bit of Port-1)

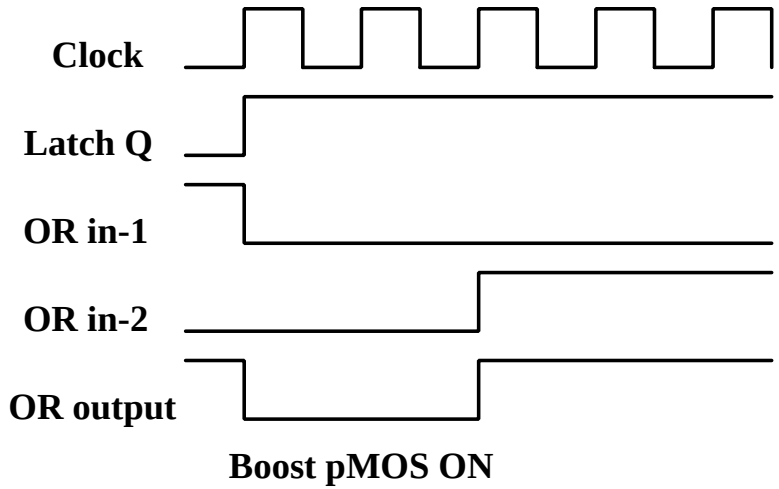
- One input to the OR gate is the inverted output of the latch bit $\overline{Q_i}$. The other input is a delayed and inverted version of $\overline{Q_i}$.
- Since the OR is fed by complementary inputs, one of its inputs must always be a '1' in steady state.

Thus in steady state, The OR output is '1' and the boost pMOS MP2 is OFF.

Port-1 driver: Boost circuit



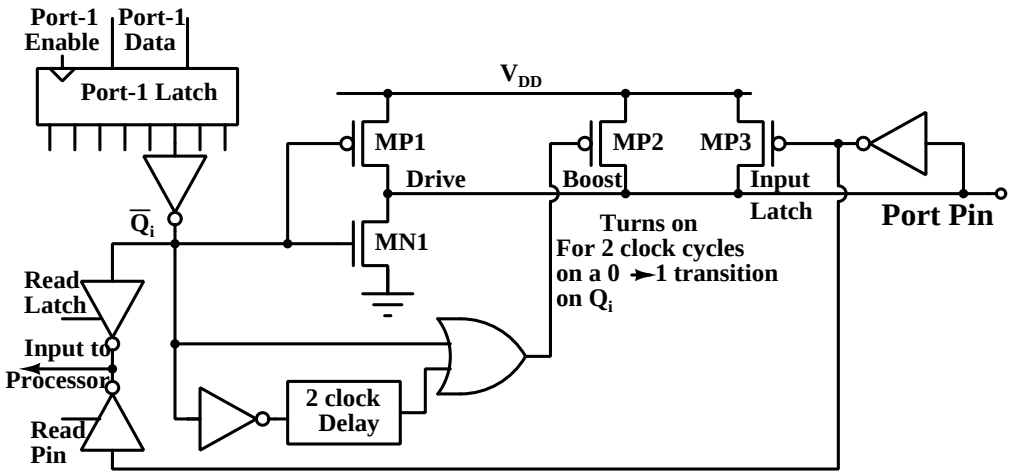
Port-1 Driver (There is one such for every bit of Port-1)



On a $0 \rightarrow 1$ transistion on port 1 latch output, the inverted output $\overline{Q_i}$ to OR gate goes from 1 to 0 immediately, but the other input goes from $0 \rightarrow 1$ only after 2 clock cycles.

So the OR output is '0' for 2 clock cycles and the strong boost transistor comes on for 2 cycles to charge the pin quickly to '1'.

Port-1 driver: Latch



Port-1 Driver (There is one such for every bit of Port-1)

- As the pin charges to ‘1’ (through the boost transistor MP2), output of the inverter connected to the pin drops to ‘0’.
- This turns on the weak pMOS MP3, which forms a latch with the input inverter.

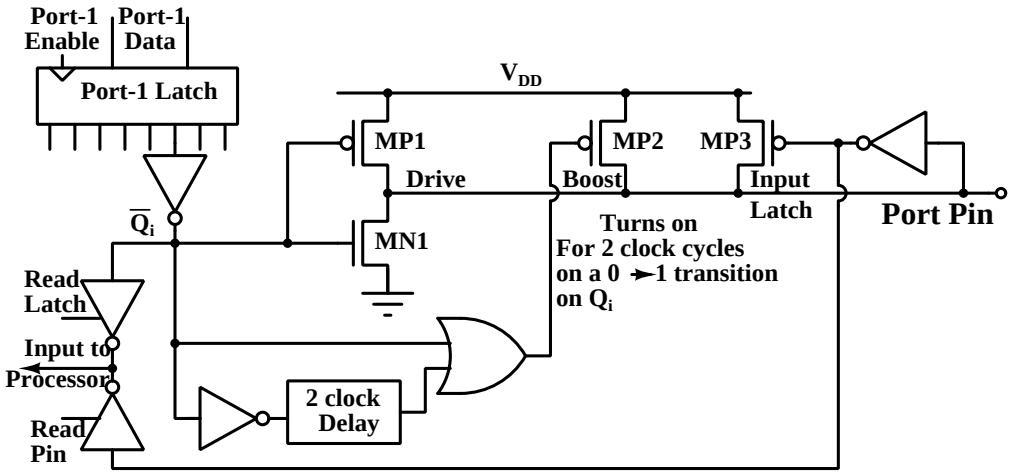
Thus the output quickly charges to ’1’ through the boost transistor, but after 2 clock cycles, the strong transistor goes OFF and the pin is maintained at ’1’ by the weak pMOS MP3.

Port-1 driver

- To use the port as an output, we just write the data to address 90H using direct addressing.
- This data appears at the port pins, buffered by two inverters.
- To use the port as an input, we write FFH to the latch. All port pins are quickly charged to ‘1’ by the boost pMOS. After 2 clock cycles, the boost pMOS goes off and the pins are pulled up only by a weak pMOS.
- The external circuit can now drive the pins to ’0’ by sinking the small amount of current sourced by the weak pMOS transistors.

Port-1 driver

The external circuit may pull the pin low, though the latch output is ‘1’.



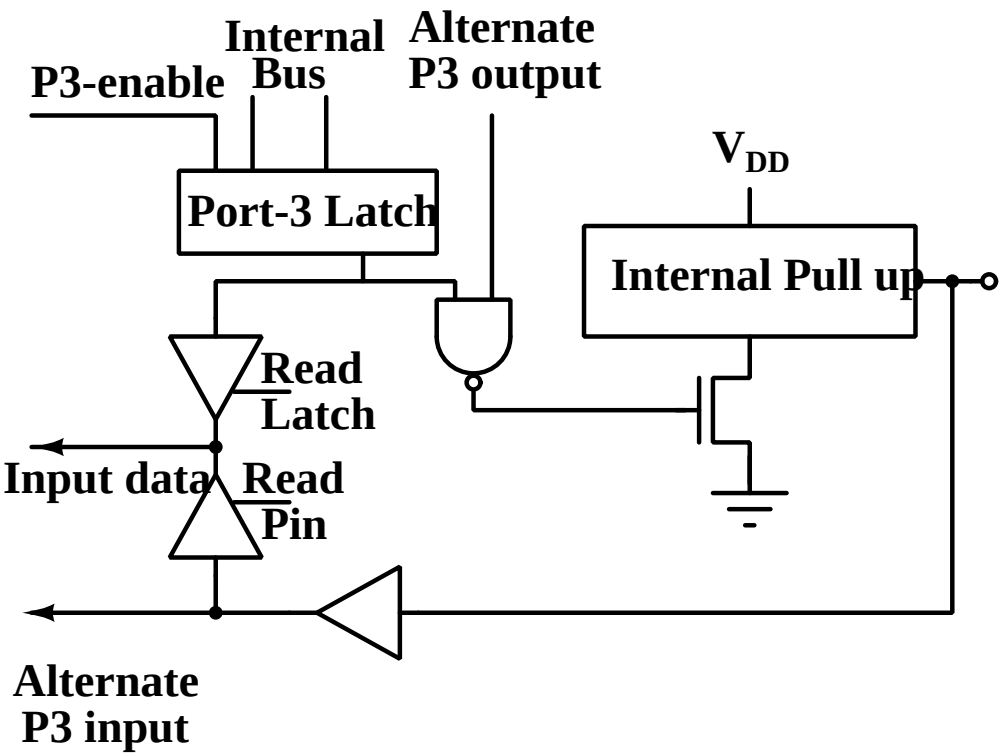
Port-1 Driver (There is one such for every bit of Port-1)

- While reading the port, the processor has to choose whether to read the ‘1’ from Latch, or the ‘0’ from the pin.
- It reads the latch if the read is a part of a Read-Modify-Write operation.
- For other read operations, the processor reads the pin, not the latch.

The arrangement for reading the latch for Read-Modify-Write operations and the pin for other input operations is implemented for all ports.

Port-3 driver

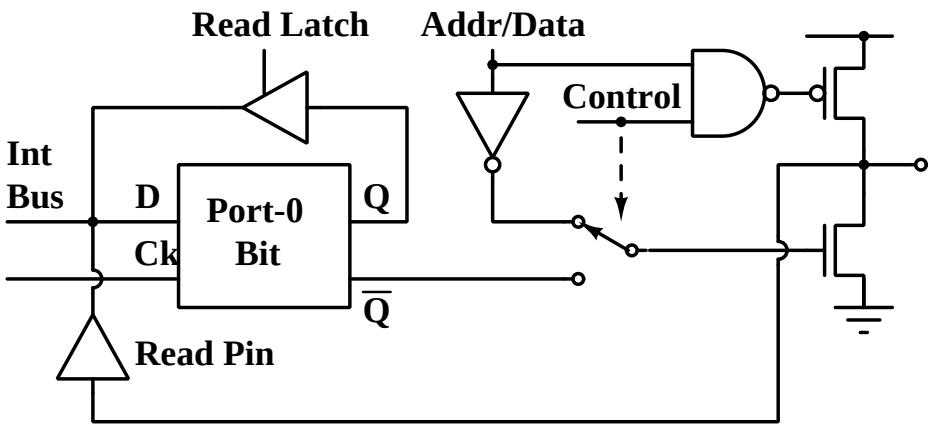
Port-3 drive is similar to Port-1 driver. However, port-3 pins may have to provide alternate functions.



- The internal pull up is the same boost and latch arrangement as for port-1.
- However, depending on the instruction being processed, the pin may be driven to an alternate value other than the latch Q output. (Say to $\overline{Rd}$ or $\overline{Wr}$).
- Similarly, the input may be sampled by alternate circuits, for example by the timer/counter.

Port-0 Driver

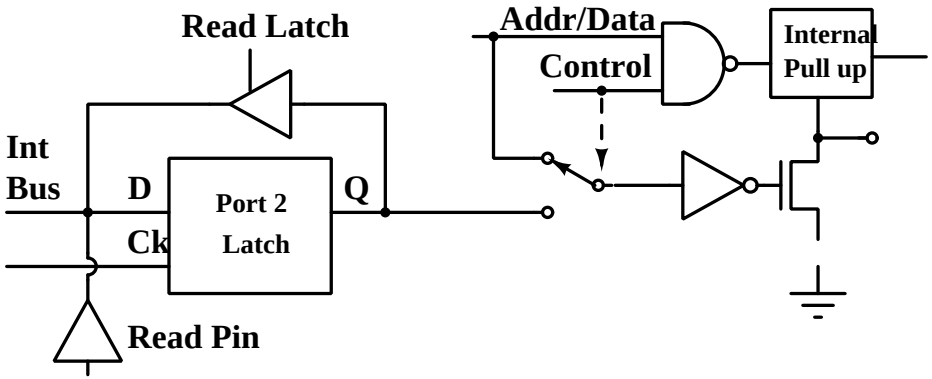
Port 0 can also be used for multiplexed address/data bus for external memory.



- Control input chooses whether the port latch or bus data will be driving the pin.
- When control is ‘0’, the port latch is selected and there is no pull up.
- When control is ‘1’, the pins are used for the external bus carrying multiplexed address and data.

Port-2 Driver

Port 2 can also be used for the upper address bytes for the external bus. When used for the external bus, these pins drive the more significant byte of the address. These are not multiplexed with data and are not required to be bi-directional.



- Control input chooses whether the port latch or bus data will be driving the pin.
- This port provides a pull up similar to port-1.

Read Modify Write operations

When the source as well as destination of an instruction is a port, this is a read modify write operation. The read part of this operation will be performed from the port register and not from the pin.
For example,

```
ANL P1, A
ORL P2, A
XRL P3, A
```

```
SET P2.5
CLR P0.3
CPL P3.5
MOV P1.2, C
```

```
INC P2
DEC P3
JBC P1.1, label
DJNZ P1, Label
```

Bit oriented operations like SET, CLR and MOV seem to be just write operations. However, these are performed by reading the full byte, masking the write operation so that it takes place only on the addressed bit and then writing back the whole byte. Therefore these are read-modify-write operations.