Phys 123 Micro 2 (Lab 20) Notes: Assembly Language; Simple I/O

REV 0; March 12, 2015.

Topics:

- What we’re after, today: giving I/O to the computer; meeting assembly language (very-low-level coding)
- address “decoding”
  - 8051 I/O decoding: done with gates or PAL (last time)
  - done with an IC decoder (74HC139)
- I/O is much simpler using built-in controller ports: keypad to display example
  - lab µ1’s little test loop (step-by-step)
  - code to use the I/O decoding done on the lab micro: keypad to display
    * Big Board Code:
      - Data Pointer (DPTR) used to define the address, on almost every off-chip data operation
      - Accumulator (A register) is the ONLY possible source or destination, for such operations
    * SiLabs Controller Code:
      - much more permissive: just move from port to port: mov p1, p0, for example

- subroutine

(class.mu2.daily.mar14.tex; March 12, 2015)
I/O Decode for 8051

This week we did last time:
Decoding with a gate

**Address:** 8000h

\[ \overline{WR} \]

**Output:**

\[ \overline{RD} \]

\[ \overline{WR} \]

\[ \overline{RD} \]

**Inputs:**

\[ \overline{AD}, A_7, A_6, A_5, A_4, A_3, A_2, A_1, A_0 \]

Binary: 1000 0000 0000 0000

Hex: 8 0 0 0

An input enables

\[ \overline{RD} \]

\[ \overline{WR} \]

\[ \overline{RD} \]

\[ \overline{WR} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]

\[ \overline{RD} \]
use decoding...

assembly language

Decimal

Mnemonic → mov dptr, #S000h; point to keypad

Source "Immediate"

Immediate

Destination → movx a, @dptr; pick up key value

R0

movx @dptr, a; send to display

S000

sujmp FOREVER; ... forever

Just peculiarity: almost always uses dedicated 16-bit pointer register to define address

For off-chip transfers (usually buses)
tiny loop 064

Address

00 80 ]
01 0E ]
02

demuxing address & data

\[ \text{ADQ}_{\text{ Reads}} = A \times D \]

\[ \text{ALE} \]

\[ \text{[catch & low] ADDRESS LINES} \]

single step

hardware \[ \rightarrow \text{CLUC} \]

just 2 clocks,
when "single stepping"