Experiment No. 7
PARALLEL INTERFACING USING THE PERIPHERAL INTERFACE ADAPTER (PIA)
ECE 441

Peter Chinetti

November 21, 2013

Date Performed:	November 14, 2013
Partners:	Zelin Wu
Instructor:	Professor Saniie

1 Introduction

1.1 Purpose

The purpose of this experiment is to introduce the student to the following topics:

- the Peripheral Interface Adapter IC (PIA, MC6821)
- the MC68000s Synchronous Bus Cycle
- the MC68000s Interrupt Generation Mechanism

1.2 Background

1.3 The Peripheral Interface Adapter (PIA)

The Peripheral Interface Adapter or PIA, provides a general purpose means of interfacing peripheral equipment to the MC68000 and MC6800 family of microprocessors. This integrated circuit is capable of interfacing a microprocessor to peripheral devices through two 8-bit bi-directional peripheral data buses (PA0 to PA7, and PB0 to PB7) and four control lines (CA1, CA2, CB1, and CB2). The functional configuration of the PIA is programmable by the CPU during system initialization. Each of the sixteen peripheral data lines can be programmed to perform either as an input or an output line. Each of the peripheral control lines may be programmed to operate in one of several modes. The speed and ease of programmability provides a high degree of flexibility for the interface.

The PIA occupies four consecutive locations in the CPUs memory map. The PIA contains six internal registers, three for Port A, and three for Port B. The two Register Select lines (RS1, RS0), are used to select one of the four registers inside the PIA.

1.4 MC6800 Synchronous Bus Cycle

In order for the 68000 to interface to 6800 type peripherals, the 68000 modifies its bus cycle to meet the 6800 bus cycle timing requirements whenever a 6800 type device is selected. This feature is possible because both types of processors use memory-mapped I/O.
The 6800 interface is provided by three signals on the 68000. These signals are known as: Enable (E), Valid Peripheral Address (*VPA), and Valid Memory Address (*VMA). *VPA and *VMA are active-low signals.

2 Lab Procedure and Equipment List

2.1 Equipment

- SANPER System
- Computer with TUTOR software

2.2 Procedure

Design and implement PIA circuit, then test with a test program.

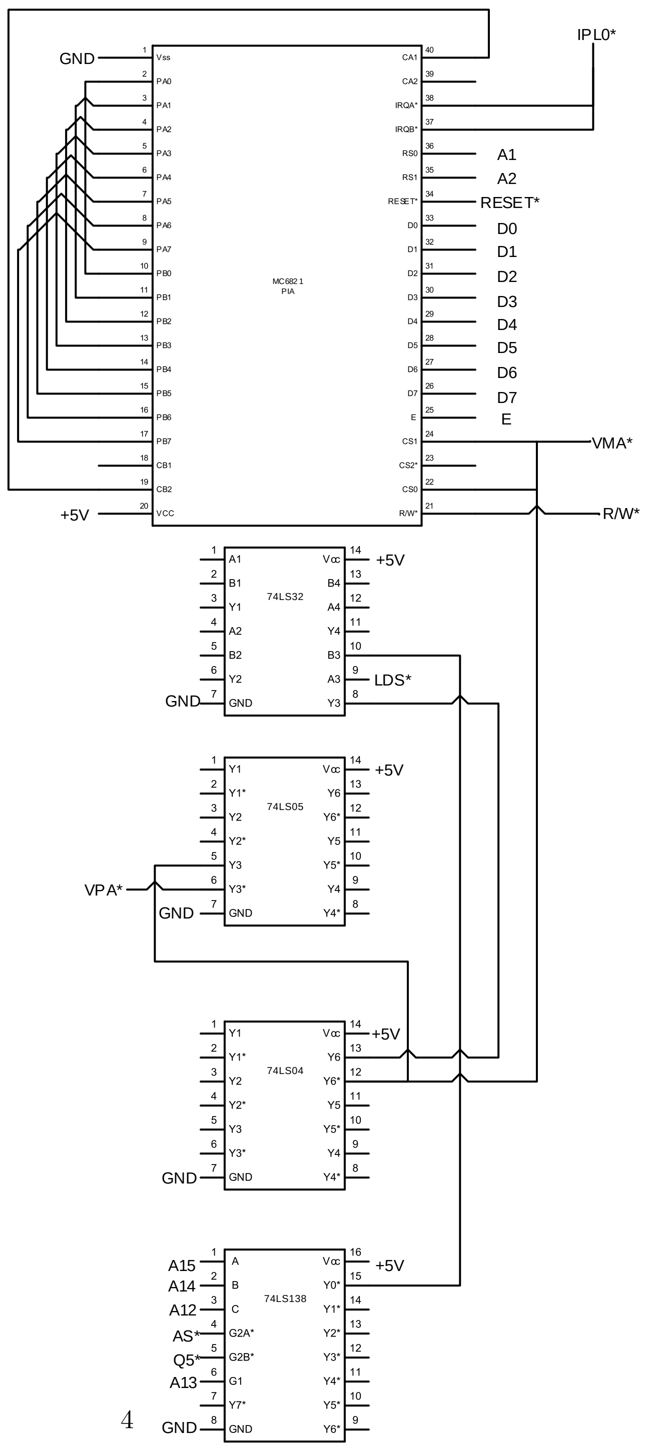
3 Results, Analysis and Discussion

3.1 Test Program

```
  ORG $100

    DC.L ISR
CRA   EQU $52003
CRB   EQU $52007
DDRA EQU $52001
DDRB EQU $52005
PDRA EQU $52001
PDRB EQU $52005

      ORG $900

  BCLR.B #2,CRA     ;set to choose DDRA
  BCLR.B #2,CRB     ;set to choose DDRB

  MOVE.B #00,DDRA     ;set PA0 to PA7 as inputs
  MOVE.B #$FF,DDRB   ;set PB0 to PB7 as outputs
  BSET.B #0,CRA    ;enable CA1 interrupt
  BSET.B #3,CRB    ;set CRB to pulse mode 101
  BCLR.B #4,CRB
  BSET.B #5,CRB

  BSET.B #2,CRA    ;set to choose PDRA
  BSET.B #2,CRB    ;Set to choose PDRB

  MOVE.B  #$07,CRA
  MOVE.B  #$2C,CRB
  MOVE.B  $52001,D0
  ANDI.W  #$F8FF,SR ;set interrupt level to 0

LOOP
    MOVE.B #247,D7       ;input single character
    TRAP #14
  MOVE.B D0,PDRB
  BRA LOOP

  ORG $1800

ISR MOVEM.L D0-D7/A0-A6,-(A7) ;store the regitsers to the stack
  MOVE.B PDRA,D0
  MOVE.B #248,D7               ;output single character
  TRAP #14
  MOVEM.L (A7)+,D0-D7/A0-A6 ;restore the registers
  RTE                       ;return from exception
```

LAB7.X68

3.2 Diagram



MC6821 PIA
74LS32
74LS05
74LS04
74LS138
GND
+5V
IPL0*
A1
A2
RESET*
D0
D1
D2
D3
D4
D5
D6
D7
E
VMA*
R/W*
LDS*
VPA*
A15
A14
A12
AS*
Q5*
A13

3.3 Program descripton

3.3.1 ISR: lines # 40-45

Read data from PIA, output it over the serial port.

3.3.2 Initilization: lines # 13 - 30

Set internal configuration of the PIA, see comments for line by line description of the settings.

3.3.3 Loop: lines # 32 - 36

Read data in from keyboard, push it out to the PIA.

3.4 Interrupt Acknowledge Method

The PIA used Auto-Vectored Interrupts because it cannot generate a vector.

3.5 Which Side?

The microprocessor has to poll the status registers of both sides to determine which generated a interruput request.

3.6 CRA=$3F

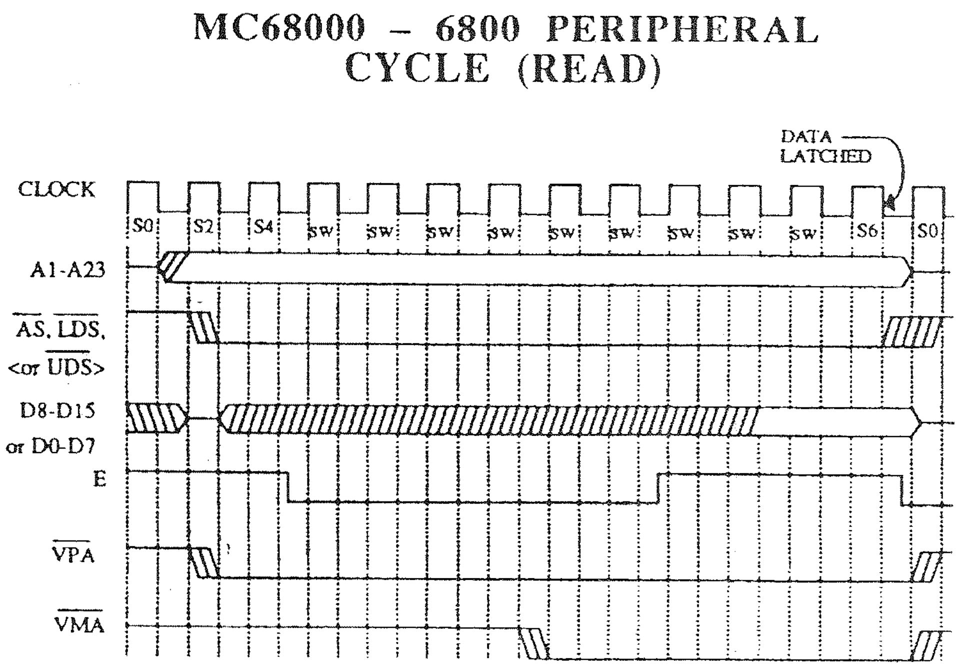
IRQA1	IRQA2	CA2 Control			DDRA	CA1 Control	
0	0	1	1	1	1	1	1

There are no pending interrupts. CA2 Control: In this mode, the CA2 output line will always be in the high state. DDRA: PDRA selected. CA1 Control, Rising transisitions create an unmasked IRQA1 interrupt.

3.7 Applications

The PIA could be used for connecting to a speedometer with an asynchronous update rate, or as an input to a motor controller, or both.

3.8 Synchronous Bus Read Cycle



a. Address lines stablize

b. VPA and (AS,LDS,or UDS) stablize.

c. Many cycles later: VMA stablizes.

d. Many cycles later: Data lines stablize.

e. A few cycles later: Data latched.

f. All lines reset.

4 Conclusions

This experiment was accomplished. An I/0 device was built and tested.