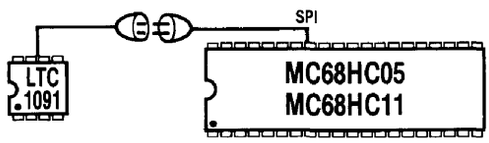
Interfacing the LTC1091 to the MC68HC05 MCU



Guy Hoover
William Rempfer

Introduction

This application note describes an interface between the LTC1091 10-bit data acquisition system and the Motorola SPI family of single chip microcomputers (e.g., 68HC05). The simple four wire interface is capable of completing a 10-bit conversion and shifting the data to the 68HC05 in 58μs. Configuration of the LTC1091 and the 68HC05 will be discussed as it applies to this interface. Schematics, code, and timing diagrams will be shown. Finally, a summary of the key points of this interface will be given, including data throughput rates.

Interface Details

The LTC1091 has one clock line which controls the A/D conversion rate and the data shift rate. Data is transferred in a half duplex, synchronous format over D_{IN} and D_{OUT}.

The Motorola Serial Peripheral Interface (SPI) is a synchronous, full duplex, serial port built into the 68HC05 that allows the user to construct a simple communication path to the LTC1091. SPI provides clock, data in and data out lines that are compatible with the LTC1091. The only additional line required is one programmable output pin (C0) to control $\overline{CS}$ on the LTC1091. The schematic of Figure 1 shows how the two devices are connected.

Hardware Description

The 68HC05 was emulated and the code for this interface was developed on a Motorola M68HC05 EVM.

$\overline{SS}$ (Pin 34) of the 68HC05 must be held high to enable the SPI properly for this interface.

The timing diagram of Figure 2 was obtained with an HP1631A logic analyzer using a 4MHz clock for the 68HC05.

The analog section of the schematic in Figure 1 is omitted for clarity. For a complete discussion of the analog considerations involved in using the LTC1091 please see the data sheet.

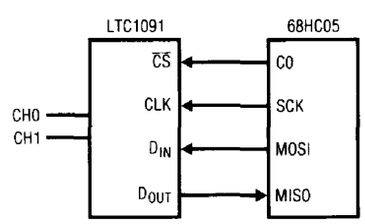


Figure 1. Schematic

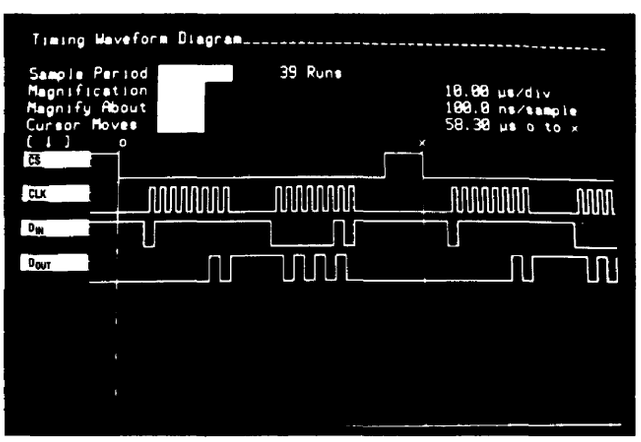


Figure 2. Timing Diagram

Software Description

The software configures and controls the SPI of the 68HC05. Additionally, the software manipulates C0 ($\overline{CS}$ of the LTC1091).

The code first configures the Serial Peripheral Control Register (SPCR) of the SPI. The SPI interrupt is disabled. The SPI outputs are enabled. The SPI is configured as a master. Finally, the SPI clock is set to normally low, for data transfer on the rising edge and for a frequency equal to one fourth the internal processor clock (one eighth the crystal frequency).

Port C is configured as all outputs by placing ones in the data direction register of port C. A D_{IN} word that configures the LTC1091 for CH1 with respect to ground and MSB first is stored in $50. Figure 3 shows how the D_{IN} word is composed. Leading zeroes in the D_{IN} word are ignored. This makes it easy to position the D_{OUT} word on exact byte boundaries so that shifting the data to right justify it is not necessary.

C0 is made to go low. D_{IN} for the LTC1091 is loaded into the SPI data register. Storing D_{IN} in the data register causes the transfer to begin. The status register of the SPI is tested until the SPIF bit is set which indicates the transfer is finished. Reading the SPI status register clears the SPIF bit and allows the data register to be read, which is the next step. The first eight bits containing the MSBs from the LTC1091 are then stored in $60 of the 68HC05 as shown in Figure 4. The LSBs are transferred in the same manner and stored in $61 of the 68HC05.

0	1	1	1	1	1	1	1
Ignore	Start	S/D	O/S	MSBF	don't care	don't care	don't care

Figure 3. 4-Bit D_{IN} Word for LTC1091 in $50

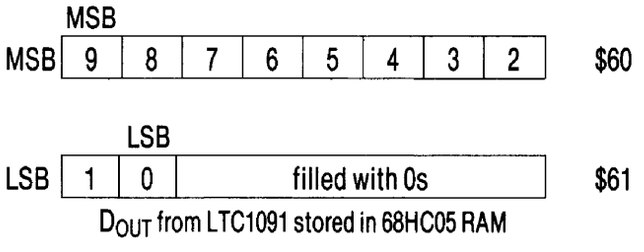


Figure 4. Memory Map

LABEL	MNEMONIC		COMMENTS
	LDA	#$51	DATA FOR SPCR
	STA	$0A	LOAD DATA INTO SPCR
	LDA	#$FF	DATA FOR DDR
	STA	$06	CONFIGURE PORT C DDR
	LDA	#$7F	D_{IN} WORD FOR LTC1091
	STA	$50	PUT D_{IN} WORD IN $50
START	BCLR	0, $02	C0 ($\overline{CS}$) GOES LOW
	LDA	$50	PUT D_{IN} WORD IN ACC
	STA	$0C	START TRANSFER
TEST	TST	$0B	TEST IF DONE
	BPL	TEST	IF NOT TRY AGAIN
	LDA	$0C	LOAD MSBs IN ACC
	STA	$0C	START NEXT TRANSFER
	AND	#$03	MASK UNUSED BITS
	STA	$60	STORE MSBs IN $60
TEST1	TST	$0B	TEST IF DONE
	BPL	TEST1	IF NOT TRY AGAIN
	BSET	0, $02	C0 ($\overline{CS}$) GOES HIGH
	LDA	$0C	PUT LSBs IN ACC
	STA	$61	PUT LSBs IN $61

Figure 5. 68HC05 Code

The code was written for the 68HC05. By changing the addresses of the special function registers however, the code should run on all of Motorola's SPI processors including the 68HC11.

Summary

A four wire interface between the LTC1091 and the 68HC05 with a combined data conversion and transfer time of 58μs was demonstrated. The interface used the serial (SPI) port of the 68HC05. The 10 data bits of the LTC1091 are shifted MSB first in two eight bit transfers. The data is stored left justified in the 68HC05's internal RAM.