I2C Protocol Design for Reusability

Zheng-wei HU
North China Electric Power University, Baoding, China
hzwwizard@hotmail.com

***Abstrac t*—One I2C protocol design method for reusability was proposed. In this method, design was divided into 3 levels: protocol level, signal level and interface level. Protocol level can be reused without any modification. Signal level can be reused by setting the number of be transferred byte according to specific operation. Interface level can be reused by changing the number of operation mode and the duration of every mode. Interface level is more complicated for reusability because different device has different number of operation mode. For low power reason, not all functions of I2C were designed. This design method was design in VHDL, implementation in FPGA and applied in bio-logging design for RTC and light sensor which are based on I2C protocol. The data acquired by light sensor were transferred through RS232 to PC and stored into text file . The file was shown into graph by using Matlab. The data acquired by RTC were shown by RS232 tool. The correctness efficiency can be confirmed by simulation results.**

Key words-I2C protocol, reusability, light sensor, RTC, bio-logging

Ⅰ. INTRODUCNTION

I2C is one popular serial data transfer protocol and used widely. I2C protocol has many merits such as controllability, less wire connection and so on[1]. I2C has several operation modes but to specific application only parts of functions will be useful. If all functions were included, the cost and power dissipation are increase. Bio-logging is battery based so low power design is necessary[2]. For the controllability, interface of each device is designed dedicatedly. I2C protocol also can be shared by more than one device. The principle is the same.

Ⅱ. LEVEL DIVISION

For the reason of reusability, I2C protocol design was divided into 3 levels in this method, As described in figure 1, these 3 levels from lowest to highest are: protocol level(PRL), signal level(SIL) and interface level(INL).

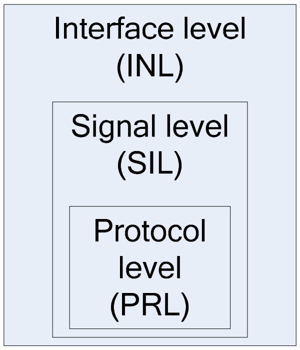


Figure 1. Level division of I2C protocol design

A. Protocol level (PRL)

Protocol level is the lowest level of all three levels. In this level, the design was performed according to the protocol control signals and timing. Writing and reading are two basic operations in this level. For reusability, a division method based byte is proposed and showed in figure 2.

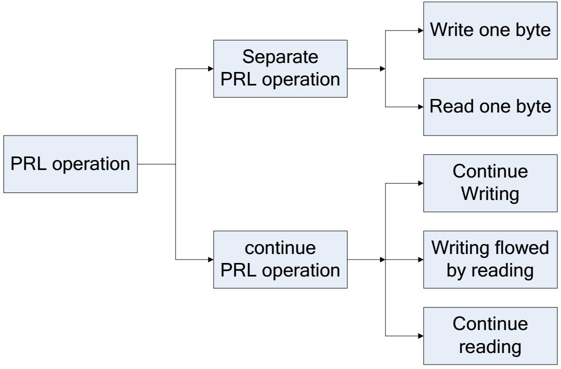


Figure 2. Description of PRL operation

According to the number of byte read or written in one pair of start-stop period, PRL operation can be divided into two types: separate PRL operation and continue PRL operation. There is only one byte was transferred during one pair of start-stop period in separate operation mode. As contrast, there are at least two bytes were transferred during one pair start-stop period in continue operation mode. In real application, continue PRL operation is widely used. Continue PRL operation includes three types: continue writing, continue reading and writing flowed by reading. For some devices whose register address can be added automatically after one read operation, the continue mode is very efficient.

In this method, the timing information is as follows:

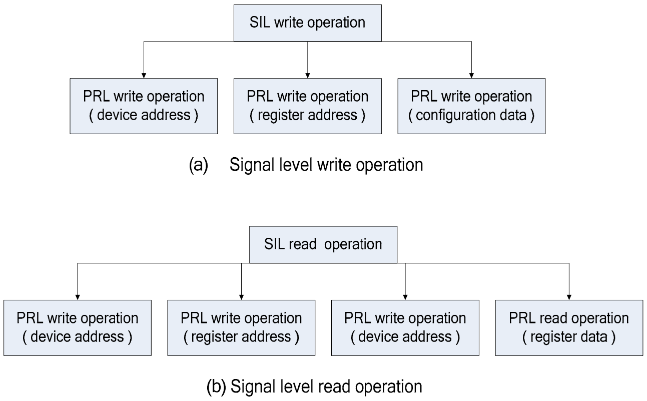
- One separate PRL operation: 17*clock_cycle
- One continue PRL operation: (N bytes were transferred)
  [17+(N-1)*(2+16)]*clock_cycle
- Interval between two PRL operation: 5*clock_cycle
- Interval between writing and reading signal in writing flowed by reading mode: 2*clock_cycle
- Clock_cycle is the period of the system clock.

B. Signal level

Signal level is the middle level of all three levels. According to the type of PRL operation, control signals were generated to realize relevant PRL operation. For example, in this level, write or read signal was used as control signal to generated start and stop signal in PRL. If write or read signal changes to 1 from 0 and SDA is 1, that is has a rising edge, then in PRL 'start' signal was generated. When write or read signal changes

978-0-7695-4261-4/10 $26.00 © 2010 IEEE
DOI

to 0 from 1 and SDA is 1, that is has a falling edge, then in protocol level 'stop' signal was generated. In order to write and read one byte correctly, the valid duration of write or read signal in separate operation mode is 17*clock_cycle. And there should be 5*clock_cycle wait time between two separated operations. For continue operations the valid duration of read or write signal is [17+(N-1)*(2+16)]*clock_cycle. N is the number of bytes were transferred. If writing m bytes flowed by reading n bytes, the during of write signal is [17+(n-1)*(2+16)]*clock_cycle and the duration of read signal is [16+(m-1)*(2+16)] *clock_cycle. The read signal should valid after 2*clock_cycle when write signal becomes invalid. The whole operation time is also [17+(n+m-1)*(2+16)]*clock_cycle. The mapping relationship between SIL and PRL was showed in figure 3. in figure 3, the PRL operations were not classified into separate operation or continue operation. One SIL write operation can mapping into three separate PRL write operations or one continue PRL operation. This is decided by the specific protocol.


SIL write operation
PRL write operation (device address)
PRL write operation (register address)
PRL write operation (configuration data)
SIL read operation
PRL write operation (device address)
PRL write operation (register address)
PRL write operation (device address)
PRL read operation (register data)

(a) Signal level write operation

(b) Signal level read operation

Figure 3. Mapping relationship between SIL and PRL

C. Interface level

Interface level is the highest level of all three levels. Design of this level was determined by special device. One device can be work in different mode according to system requirement. One device can be in work mode or low power mode.

In this level, the main design task is state machine design. The most important issue is timing of every state. The design method can be showed by one application of light sensor in section A of Ⅳ.

Ⅲ. DATA TRANSFORMATION

A. Written data generation

In order to realize some functions such as recognize the device and registers inside of it, the operation is writing or reading and so on, the master should send written data to I2C and received by slave.

In this method, written data generation was realized in protocol level. There is one state for generating written data and loading the written data into one shift register.

B. Data shift

One byte was transferred as one unit, so one shift register was needed to realize transformation between parallel data and serial data. For write operation, parallel data were transformed into sequence data. For read operation, sequence data were transformed into parallel data.

C. Data readout

There should be one signal to indicate the validation of read data. It can also be as the write enable signal for SD card to store the read data.

Ⅳ. APPLICATIONS

According to this design method, two applications were realized. One is for light sensor and the other is for RTC. This method was design in VHDL and implemented in FPGA.

A. Light sensor (ISL29015)[3]

a) Reading operation

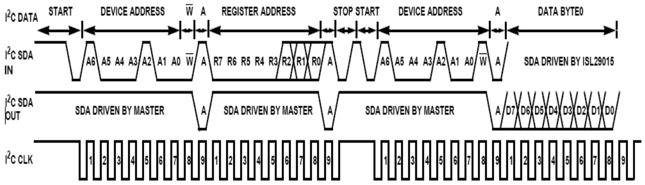


Figure 4 Timing of read operation

- Signal level: one SIL read operation
- Protocol level: two continued PRL operations. The first continue PRL operation is continue writing and the second continue PRL operation is writing flowed by reading.
- Two bytes were written in the first continue PRL operation, so the duration is : [17+(2-1)*(2+16)]*clock_cycle.
- One byte was written and one byte was read in the second continue PRL operation, so the duration is: [17+(2-1)*(2+16)]*clock_cycle.
- The duration of one SIL operation is (17+2+16)+5+(17+2+16)=75 clock cycles.

b) Writing operation

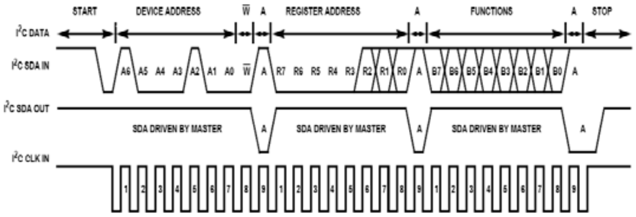


Figure 5 Timing of write operation

- Signal level: one SIL operation

- Protocol level: one continue PRL writing operation. This PRL operation includes three continued writing operations
- The duration of one SIL operation is [17+(3-1)*(2+16)]*clock_cycle=53 clock_cycle

c) *Interface level design*

The design of interface level is depend on specific device. In this application, the type of light sensor is ISL29015. The light sensor was used to acquire the light data of the environment. The working process of light sensor can be showed as figure 6. Because different device has different number of registers to be configured, the value of configuration is different. There is the same reason as value of readout. The number of state is also different but can be added or deleted according to specific device.

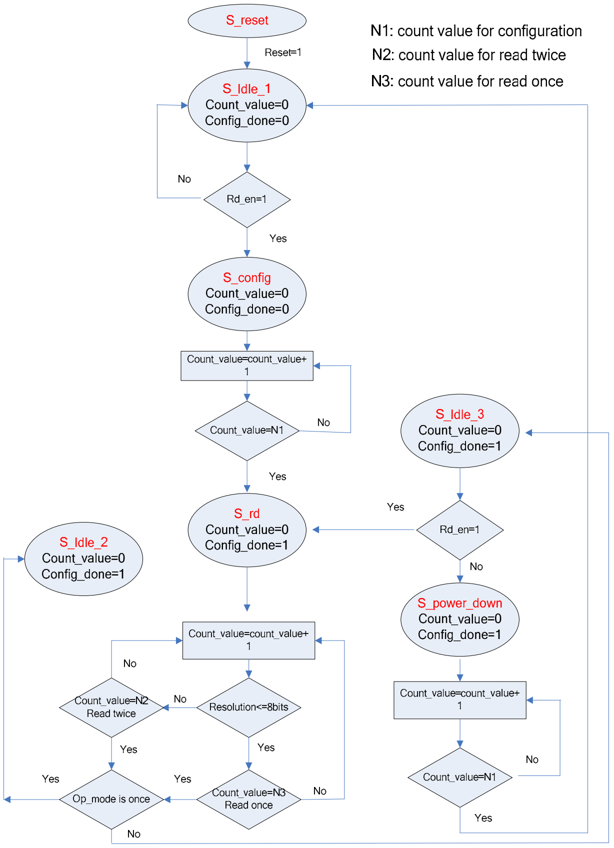


Figure 6 interface level control state machine

d) *Written data generation*

Data registers store the write data for each SIL operation. In different mode, SIL operation includes different number of PRL writing operations. For this application, there are three state of light sensor, and the number of PRL writing operations is different. This can be showed by figure 7.

B. RTC(M41T62)[4]

a) *Writing operation*

- SIL operation: one SIL writing operation
- PRL operation: one continue RPL writing operation. In one continue PRL operation 17 bytes were written (device address + register address+15 configuration data).
- The duration of SIL writing operation is: [17+(17-1)*(2+16)]*clock_cycle=305 clock_cycles

b) *Reading operation*

- SIL operation: one SIL reading operation
- PRL operation: two continue PRL operation. The first PRL operation is continue writing and the second is writing followed by reading.
- Two bytes were written in the first continue PRL operation, so the duration is : [17+(2-1)*(2+16)]*clock_cycle.
- One byte was written and seven bytes were read in the second continue PRL operation. The duration is: [17+(8-1)*(2+16)]*clock_cycle.
- The duration of one SIL operation is: [17+(2-1)*(2+16)]+5+[17+(8-1)*(2+16)]=183 clock_cycles

c) *interface level design*

The same as light sensor only different in the number of state and the counter value.

d) *write data generation*

The same as light sensor only different in the number of registers.

000	Device address for write
001	Reg1__adr(00h)
010	Reg1__commamd
011	Reg2__adr(01h)
100	Reg2__commamd
101	Reg3__adr(02h)
110	Device address for read
111	Reg4__adr(03h)

(a) Register architecture

000	Device address for write
001	Reg1__adr(00h)
010	Reg1__commamd
000	Device address for write
011	Reg2__adr(01h)
100	Reg2__commamd

(b) Write data for configuration

000	Device address for write
101	Reg3__adr(02h)
110	Device address for read
000	Device address for write
111	Reg4__adr(03h)
110	Device address for read

(c) Write data for read

000	Device address for write
001	Reg1__adr(00h)
001	Reg1__commamd(00h)
000	Device address for write
011	Reg2__adr(01h)
001	Reg2__commamd(00h)

(d) Write data for power down

Figure 7 data register

Ⅴ. SIMULATION AND RESULTS ANALYSIS

Figure 8 shows the results of light sensor in the open circumstance for 3 days. Figure 9 shows the results of light sensor in dark box for one day and the value is unchanged because the system was in one dark box. Figure 10 is the results of RTC. Form left to right the values are: second, minute, hour, week, date, month, year. In the data of week, there are some bits for control. The system was working in 32 KHz. Light sensor and RTC were read every second.

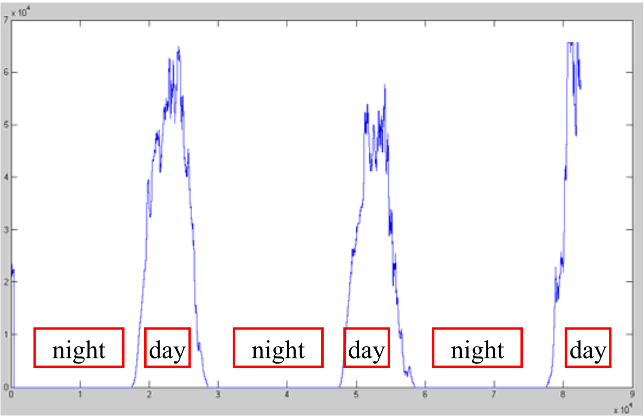


Figure 8 light sensor acquisition data in open circumstance

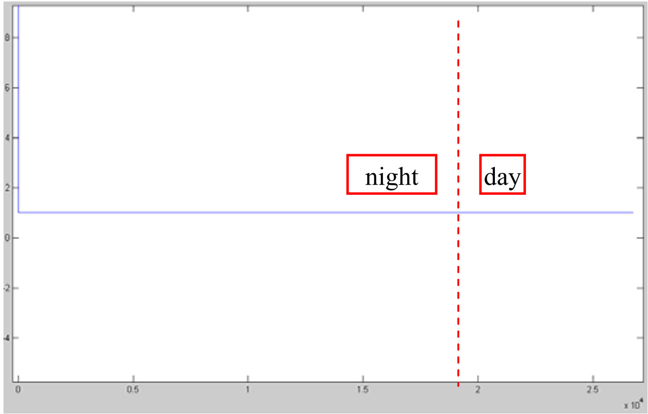
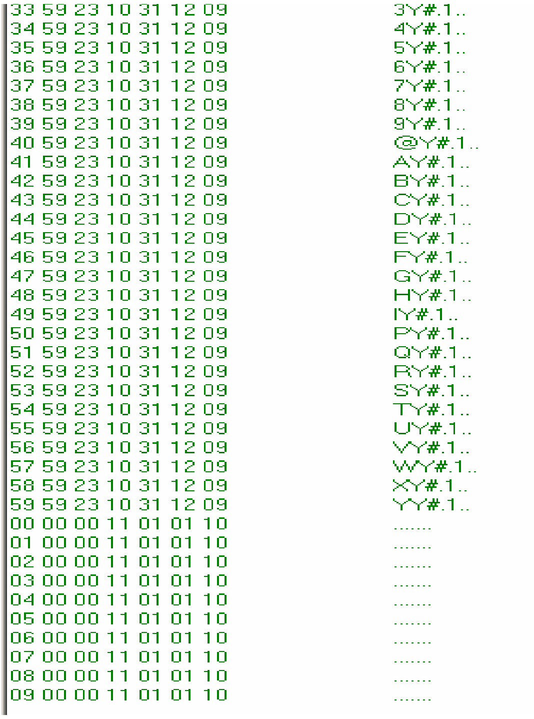


Figure 9 light sensor acquisition data in dark box



```
33 59 23 10 31 12 09                3Y#.1..
34 59 23 10 31 12 09                4Y#.1..
35 59 23 10 31 12 09                5Y#.1..
36 59 23 10 31 12 09                6Y#.1..
37 59 23 10 31 12 09                7Y#.1..
38 59 23 10 31 12 09                8Y#.1..
39 59 23 10 31 12 09                9Y#.1..
40 59 23 10 31 12 09                @Y#.1..
41 59 23 10 31 12 09                AY#.1..
42 59 23 10 31 12 09                BY#.1..
43 59 23 10 31 12 09                CY#.1..
44 59 23 10 31 12 09                DY#.1..
45 59 23 10 31 12 09                EY#.1..
46 59 23 10 31 12 09                FY#.1..
47 59 23 10 31 12 09                GY#.1..
48 59 23 10 31 12 09                HY#.1..
49 59 23 10 31 12 09                IY#.1..
50 59 23 10 31 12 09                PY#.1..
51 59 23 10 31 12 09                QY#.1..
52 59 23 10 31 12 09                RY#.1..
53 59 23 10 31 12 09                SY#.1..
54 59 23 10 31 12 09                TY#.1..
55 59 23 10 31 12 09                UY#.1..
56 59 23 10 31 12 09                VY#.1..
57 59 23 10 31 12 09                WY#.1..
58 59 23 10 31 12 09                XY#.1..
59 59 23 10 31 12 09                YY#.1..
00 00 00 11 01 01 10                .......
01 00 00 11 01 01 10                .......
02 00 00 11 01 01 10                .......
03 00 00 11 01 01 10                .......
04 00 00 11 01 01 10                .......
05 00 00 11 01 01 10                .......
06 00 00 11 01 01 10                .......
07 00 00 11 01 01 10                .......
08 00 00 11 01 01 10                .......
09 00 00 11 01 01 10                .......
```

Figure 10 RTC readout data

Ⅵ. CONCLUSIONS

One design method of I2C protocol for reusability was proposed and applied in bio-logging design. This method was emulated by FPGA with light sensor and RTC. The simulation results indicated this method is correct and efficient.

ACKNOWLEDGMENT

Suported by youth fundation of north china electric power university through 200712001.

REFERENCES

[1]The I2C-bus specification, version 2.1,januaray 2000.
[2] G. Roberts, A. Williams. A LOW-POWER POST-PROCESSED DGPS SYSTEM FOR OCEANOGRAPHIC APPLICATIONS[J]. Electronic Engineering in oceanography. 19-21 July, 1994. pp 18-23
[3]ISL29015 datasheet[OB/DL]. www.intersil.com/data/fin/fn/6522.pdf
[4]M41T62 datasheet[OB/DL]. www.st.com/stonline/products/literature/ds/10397/m41t64.pdf