*بسم الله الرحمن الرحيم*

**

**

*Computer Systems Engineering Department*

*Digital Electronics and Organization Computer Lab*

*ENCS.211*

*Report for Experiment No.2*

*Adders, Subtractors, Comparators*

*Instruction: Dr. Iyad Tumar*

*Teaching Assistant: ENG. Anas Abd-elrazeq.*

*Students Name & ID:*

*Laith Samara, 1080758*

*Ahmad Mafarjah, 1080429*

*Date: 7.9.2010*

*Section (4)*

* ***Theory:***

1. *Comparator:*

*Comparator is a combinational circuit used to compare at least two numbers and constructed from basic gates AND, OR, X-OR… etc. and assigned the result which number is greater, less than or equal the other one by generate 0, 1.*

1. *Adders circuits:*
2. *Half-Adder:*

*Half-Adder is a combinational circuit constructed from basic gates, and used to performe addition of two bits and the half-adder assign the result by generating 2 values one is sum and the other is carry.*

1. *Full-Adder:*

*Full-Adder is a combinational circuit constructed from basic gates, and used to performe addition operation of three values one of them is the previous carry-out of the previous half-adder*

*And generates the result as two values one of them is the sum and the other is the carry.*

1. *Subtractor Circuits:*
2. *Half –Subtractor:*

*Half-Subtractor is a combinational circuit constructed from basic gates, that performe a subtraction of 1-bit, this kind of circuit doesn’t take “Borrow” from the previous subtraction under consideration and generates two outputs the first is the result of subtractions and the second is the borrow, this binary operation is performed using 2’s complement. The complement steps:*

* *The subtracted is inverted to 1’s complement by convert each 1to 0and vice versa*
* *Add “1”to the least significant bit of the subtrahend in 1’s complement.*

1. *Full-Subtractor:*

*Full-Subtractor is a combinational circuit constructed from basic gates, that performe a subtraction and consider the value of borrow from the previous stages as one of the inputs and generate two outputs, the first one is the result of subtraction and the other is the borrow*.

* ***Procedure:***

***Comparator:***

1. ***Constructing Comparator with Basic Logic Gates.***

*As a first step, we Set the KL-26001 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block a, then we completed the connections by referring to wiring diagram in Fig 1.1(a).*

**

(a)Wiring diagram (KL-26001 block a)

**

(b) Logic Diagram

*FIG 1.1: 1-bit comparator*

*The inputs were active high. We Connected inputs A and B to Data Switches SW1 and SW2 (The outputs are active low). Then we Connected outputs F1, F2, F5 to logic Indicators L1, L2, L3, respectively and we Applied +5 VDC from the Fixed Power on KL-26001 Module. We followed, the input sequences in Table 1.1. And we recorded the outputs we observed.*

|  |  |  |
| --- | --- | --- |
| *INPUTS*  *B A*  *SW2 SW1* |  | *OUTPUTS*  *F1 F2 F5*  *(L1) (L2) (L3)* |
| *0 0* | *A=B* | *1 1 0* |
| *0 1* | *A>B* | *0 1 1* |
| *1 0* | *A<B* | *1 0 1* |
| *1 1* | *A=B* | *1 1 0* |

*Table 1.1*

*We note from the last table that F5 is active when A =B, at AB + A'B' = A XOR B*

*F1 is active at AB’, and F2 is active at A’B, and fig 1.1 (b) represents the logic diagram of those functions.*

***(B) Constructing Comparator with TTL IC.***

*We Set the KL26005 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block (a), we apply +5VDC from the Fixed Power on KL-22001 Lab to KL26005 Module. U6 is a 7485 4-bit comparator IC. Its pin assignment and function table are given below:*

**

**

*FIG1.2: 26005 block a*

*We connected the inputs A1~A4 to SW4 ~ SW7 and B1 ~ B4 to SW0 ~ SW3, Respectively. And we Connected the outputs A=B to L1, A<B to L2, and A>B to L3. We followed the input sequences in Table1.2. We observe and recorded the results:*

|  |  |
| --- | --- |
| *Inputs*  *A4 A3 A2 A1 B4 B3 B2 B1*  *SW7 SW6 SW5 SW4 SW3 SW2 SW1 SW0* | *OUTPUTS*  *L3 L2 L1*  *A>B A<B A= B* |
| *0 0 0 0 0 0 0 0* | *0 0 1* |
| *0 1 0 1 0 0 1 1* | *1 0 0* |
| *0 1 0 0 1 0 0 0* | *0 1 0* |
| *1 0 1 0 1 0 1 0* | *0 0 1* |
| *0 1 1 0 1 0 1 1* | *0 1 0* |
| *1 1 1 1 1 1 0 0* | *1 0 0* |

*Table 1.2*

*1.5.2. Half- and Full-Adder Circuits*

***A. Constructing Half- and Full-Adders with Basic logic Gates***

*We Set the KL-26002 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block a. we completed the connections by referring to the wiring diagram in Fig1.2, we Applied +5VDC from Fixed*

*Power on the KL-22001 Lab to KL-26002 Module.*

FIG1.3: Wiring Diagram (KL-26002 Block a)

**

FIG1.4: Half-Adder Circuit

*We connected inputs A and B to Data Switches SW0 and SW1, respectively.*

*And we Connected output F1 and F2 to logic Indicators L1 and L2. We followed the input sequence for A and B in Table1.3 and recorded the outputs:*

|  |  |
| --- | --- |
| *Inputs*  *SW1(B) SW0(A)* | *OUTPUTS*  *CARRY(F1) SUM(F2)* |
| *0 0* | *0 0* |
| *0 1* | *0 1* |
| *1 0* | *0 1* |
| *1 1* | *1 0* |
|  |  |

*Table 1.3*

*From the previous table we can note that the summation of two numbers of one bit, is representing by XOR gate, SUM = A XOR B. the carry = A AND B, and that is shown in Fig 1.3.We completed the connections by referring to the wiring diagram in Fig1.5 and the full-adder circuit in Fig1.4.*

**

FIG1.4: Full-Adder Circuit

**

*FIG1.5: Wiring Diagram (KL-26002 Block a)*

*We Connected A, B & C to SW1, SW2, and SW3. The input A represents the augend, input B the addend, and the C is the previous carry. We connected the outputs F3 and F5 to Logic Indicators L1 and L2, respectively.*

*We followed the input sequence in Table1.4 and recorded the output states:*

|  |  |
| --- | --- |
| *INPUT*  *SW3(c ) SW2(B) SW1(A)* | *OUTPUT*  *CARRY (F3) SUM(F5)* |
| *0 0 0* | *0 0* |
| *0 0 1* | *0 1* |
| *0 1 0* | *0 1* |
| *0 1 1* | *1 0* |
| *1 0 0* | *0 1* |
| *1 0 1* | *1 0* |
| *1 1 0* | *1 0* |
| *1 1 1* | *1 1* |

*Table 1.4*

*After using the k\_map we get the function for the carry and sum we get:*

*Carry = AC + BC +AB.*

*SUM = CA'B' + C'AB' + CBA +C'BA' = C'(B'A + BA') + C (A'B' +AB)*

*= C XOR B XOR A.*

*And the diagram in Fig (1.4) shows those functions.*

1. ***Constructing 4-Bit Full-Adder with IC***

*At this part we Set the KL-26002 Module on the KL-22001 Basic Electricity Lab, and locate block b. The U5, 7483 is a 4-bit binary adder. We connected input Y5 to ground “0”, so the XOR gates of U6, which are connected to Y0~Y3, will act as buffers. Then, we Connected inputs X0~X3 (addend) and Y0~Y3 (augends) to Data Switches SW0~SW3 and SW4~SW7 respectively. We connected F1 (Carry out) to L1 and Σ0 ~ Σ3 (sum) to L2~L5. Apply +5VDC from the Fixed Power on KL-22001 Lab to KL-26002 Module. We followed input sequences in Table 1.5 and record the outputs F1 in binary and Σ in hexadecimal.*

*X=X3X2X1X0, Y=Y3Y2Y1Y0, Σ= Σ3Σ2Σ1Σ0.*

**

*FIG1.6: Wiring Diagram (KL-26002 block b)*

|  |  |
| --- | --- |
| *INPUT*  *X Y* | *OUTPU*  *F* |
| *0000 0000*  *0000 0001*  *0000 0110*  *0000 1001*  *0000 1111*  *0001 0011*  *0001 0110*  *0001 1000*  *0011 0110*  *0100 1000*  *0100 1111*  *1000 0111*  *1001 1001*  *1010 1011*  *1100 1110*  *1111 1111* | *0000 0*  *0001 0*  *0110 0*  *1001 0*  *1111 0*  *0100 0*  *0111 0*  *1001 0*  *1001 0*  *1100 0*  *0011 1*  *1111 0*  *0010 1*  *0101 1*  *1010 1*  *1110 1* |

*Table 1.5*

1. ***Constructing BCD Adder***

*We set the KL-26002 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block b. The circuit, shown in Fig1.6, will act as a BCD adder. We Connected inputs X0~X3 to SW0~SW3, Y0~Y3 to SW4~SW7, Y5 to ground (“0”).U5 and U9 are 7483 4-bit binary full adder, we connected outputsF8~F11 of U5 to the inputs of one of the digital displays. F8~F11 are also connected to Logic Indicators L1~L4.*

*We Connected F1 and F2 to logic Indicators L5 and L6, respectively. We also connected outputs F4~F7 of U9 to inputs of another Digital Display. Then, we connected F4~F7 to L0~L3 and F3 to L4. F11~F8 are the sum of X0~X3 added to Y0~Y3 while F1 is the carry.*

*We followed the input sequences for X0~X3 and Y0~Y3 in the table 1.6, and record the output state.*

|  |  |  |  |
| --- | --- | --- | --- |
| *X3 X2 X1 X0* | *Y3 Y2 Y1 Y0* | *OUTPUTS (U5)*  *F1 F11  F10 F9 F8* | *FINAL*  *F2 F3 F7 F6 F5 F4* |
| *0 0 0 0* | *0 0 0 0* | *0 0 0 0 0* | *0 0 0 0 0 0* |
| *0 0 0 1* | *0 0 1 1* | *0 0 1 0 0* | *0 0 0 1 0 0* |
| *0 0 1 1* | *0 1 0 0* | *0 0 1 1 1* | *0 0 0 1 1 1* |
| *0 0 1 0* | *0 0 1 0* | *0 0 1 0 0* | *0 0 0 1 0 0* |
| *0 0 1 0* | *1 0 0 0* | *0 1 0 1 0* | *1 1 0 0 0 0* |
| *0 0 1 1* | *0 1 1 0* | *0 1 0 0 1* | *0 0 1 0 0 1* |
| *0 1 0 0* | *0 0 1 0* | *0 0 1 1 0* | *0 0 0 1 1 0* |
| *0 1 0 0* | *0 1 0 1* | *0 1 0 0 1* | *0 0 1 0 0 1* |
| *0 1 0 0* | *0 1 1 0* | *0 1 0 1 0* | *0 1 0 0 0 0* |
| *0 1 0 1* | *0 1 1 0* | *0 1 0 1 1* | *0 1 0 0 0 1* |
| *0 1 1 0* | *0 1 1 1* | *0 1 1 0 1* | *0 1 0 0 1 1* |
| *0 1 1 1* | *1 0 0 0* | *0 1 1 1 1* | *0 1 1 0 0 0* |
| *0 1 1 1* | *1 0 0 1* | *1 0 0 0 0* | *1 1 0 1 0 1* |
| *1 0 0 0* | *1 0 0 1* | *1 0 0 0 1* | *1 1 0 1 1 1* |
| *1 0 0 1* | *1 0 0 1* | *1 0 0 1 0* | *1 1 1 0 0 0* |
| *1 0 1 0* | *1 0 1 0* | *1 0 1 0 0* | *1 1 1 0 1 0* |
| *1 0 1 0* | *1 0 1 1* | *1 0 1 0 1* | *1 1 1 0 1 1* |
| *1 0 1 0* | *1 1 0 0* | *1 0 1 1 0* | *1 1 1 1 0 0* |
| *1 0 1 1* | *1 1 1 0* | *1 1 0 0 1* | *1 1 1 1 1 1* |
| *1 1 1 1* | *1 1 1 1* | *1 1 1 1 0* | *1 0 0 1 0 0* |

*Table (1.6)*

*After that, we set the KL-26002 Module on the KL-22001 Basic Electricity Circuit Lab, and locate block a. And complete the connections by referring to the wiring diagram in* ***Fig1.7****. Then, apply +5VDC from Fixed Power on KL-22001 Lab to KL-26002 Module.*

**

*FIG1.7: Wiring Diagram (KL-26002 block a).*

*After that, we connect inputs A~C to Data Switches SW0~SW2; outputs F2 to L1, F1 to L2, F3 to L3 & F5 to L4. And follow the input sequences as shown in Table (1.8).*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Inputs* | | | *BW1* | *Dif1* | *BW2* | *Dif2* |
| *C* | ***A*** | ***B*** | ***F1*** | ***F2*** | ***F3*** | ***F5*** |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *0* | *1* | *1* | *1* | *1* | *1* |
| *0* | *1* | *0* | *0* | *1* | *1* | *1* |
| *0* | *1* | *1* | *0* | *0* | *1* | *0* |
| *1* | *0* | *0* | *0* | *0* | *0* | *1* |
| *1* | *0* | *1* | *1* | *1* | *0* | *0* |
| *1* | *1* | *0* | *0* | *1* | *0* | *0* |
| *1* | *1* | *1* | *0* | *0* | *1* | *1* |

*Table (1.8)*

*We note that, when C=0 the circuit is half-Subtractor with the borrow output F1 (BW1), and the differences output F2 (Dif1) =A-B. And when C=1 the circuit is a full-Subtractor with borrow output F3 (BW2) and the differences output F5 (Dif2) = A-B-C.*

*After that, we set the KL-26002 Module on the KL-22001 Basic Electricity Circuit Lab, and locate* ***block b****. And complete the connections by referring to the wiring diagram in* ***Fig1.7****. Then apply +5VDC from the Fixed Power on KL-22001 lab to KL-26002 Module.*

**

*FIG1.8: Wiring Diagram (KL-26002 block b).*

*Then, we connect inputs X3~X0 to SW7~SW4; Y3~Y0 to SW3~SW0. And connect outputs F1 to L1; F11 to L5~L2. To execute the subtract operation, we connectY5 to +5V (“1”) (or Cin of U5=1). And follow the input sequences and record the output states in* ***Table 1.9.***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Inputs* | | | | | | | | *Borrow* | *Difference* | | | |
| *X3* | *X2* | *X1* | *X0* | *Y3* | *Y2* | *Y1* | *Y0* | *F1* | *F11* | *F10* | *F9* | *F8* |
| *0* | *1* | *0* | *0* | *0* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *1* | *0* | *0* | *0* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *1* |
| *1* | *0* | *0* | *0* | *0* | *0* | *1* | *1* | *0* | *0* | *1* | *0* | *1* |
| *1* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *0* | *0* | *1* | *1* | *1* |
| *1* | *0* | *0* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *1* |
| *1* | *0* | *0* | *1* | *0* | *1* | *1* | *1* | *0* | *0* | *0* | *1* | *0* |
| *1* | *0* | *1* | *0* | *0* | *1* | *1* | *0* | *0* | *0* | *1* | *0* | *0* |
| *1* | *0* | *1* | *0* | *0* | *1* | *0* | *1* | *0* | *0* | *1* | *0* | *1* |
| *1* | *0* | *1* | *1* | *1* | *0* | *1* | *0* | *0* | *0* | *0* | *0* | *1* |
| *1* | *1* | *1* | *1* | *1* | *0* | *1* | *0* | *0* | *0* | *1* | *0* | *1* |

*Table (1.9)*

* ***Problem: Binding Comparator Circuit***

*A 4-input, 3-output circuit that compares 2-bit unsigned numbers and output (1) on one of three output lines according to whether the first number is greater than, equal to, or less than the other number. You can only use two 4×1 multiplexers.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *A* | | *B* | | *A>B* | *A<B* |
| *A1* | *A0* | *B1* | *B0* | *F1* | *F2* |
| *0* | *0* | *0* | *0* | *0* | *1* |
| *0* | *0* | *0* | *1* | *0* | *1* |
| *0* | *0* | *1* | *0* | *0* | *1* |
| *0* | *0* | *1* | *1* | *0* | *0* |
| *0* | *1* | *0* | *0* | *1* | *0* |
| *0* | *1* | *0* | *1* | *0* | *1* |
| *0* | *1* | *1* | *0* | *0* | *1* |
| *0* | *1* | *1* | *1* | *0* | *0* |
| *1* | *0* | *0* | *0* | *1* | *0* |
| *1* | *0* | *0* | *1* | *1* | *0* |
| *1* | *0* | *1* | *0* | *0* | *0* |
| *1* | *0* | *1* | *1* | *0* | *1* |
| *1* | *1* | *0* | *0* | *1* | *0* |
| *1* | *1* | *0* | *1* | *1* | *0* |
| *1* | *1* | *1* | *0* | *1* | *0* |
| *1* | *1* | *1* | *1* | *0* | *0* |

*Table (1.10)*

*Take (A0, A1) are the selection lines for the two MUXes*

*🡺 When*

*A1A0= 00: (D0)*

*F1 = 0 F2 = B1'+B0'.*

|  |  |
| --- | --- |
| *0* | *0* |
| *0* | *0* |

|  |  |
| --- | --- |
| *1* | *1* |
| *1* | *0* |

*🡺 When A1A0= 01: (D1)*

|  |  |
| --- | --- |
| *1* | *0* |
| *0* | *0* |

|  |  |
| --- | --- |
| *0* | *1* |
| *1* | *0* |

*F1 = B1'B0' F2 = B1'B0 + B1B0'.*

*= B1⊕B0.*

*🡺 When A1A0= 10: (D2)*

*F1= B1'. F2=B1B0*

|  |  |
| --- | --- |
| *1* | *1* |
| *0* | *0* |

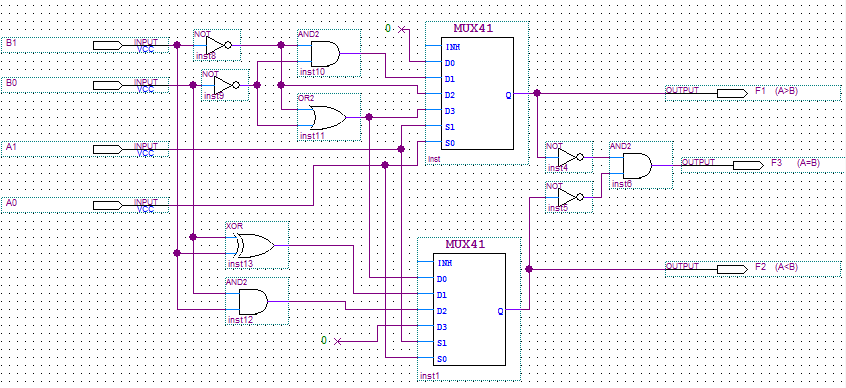
|  |  |
| --- | --- |
| *0* | *0* |
| *0* | *1* |

*🡺 When A1A0= 11: (D3)*

*F1= B1'+ B0'. F2= 0.*

|  |  |
| --- | --- |
| *1* | *1* |
| *1* | *0* |

|  |  |
| --- | --- |
| *0* | *0* |
| *0* | *0* |

**

*Fig1.9*

* ***Conclusion:***

*In this experiment ,we understand how the Full and Half Adder work and how we can construct it ,also how to build Full–Adder form two Half -Adders ,therefore we were able to construct 4-bit from Full-Adder, and in this experiment we used an IC, that IC is 4-bit Adder to understand how this adder work, also we constructed BCD Adder by using two 4-bit Adder and basic gates to Add 6 to original number when the number greater than 9,but we faced a problem and that problem when the number greater than 15 the carry must be shown , but actually not shown. I think that problem that happen since we need to add another indicator to give it to the next 4 bits used to implement the carry, therefore the carry are not shown.*

*Comparator, in this experiment we understand how 2-bit comparator work, also how 4-bit comparator work by using IC comparator. In this section of the experiment we didn’t face any problem as the other groups especially with the data input switches because I think the kit we were worked on was working probably well.*

*Subtractor, in this experiment we understand how the Half and Full Subtractor work, also how to construct Half-Subtractor from basic gates , therefore we were able to construct Full-Subtractor from Half-Subtractor . Also we constructed 4-bit Subtractor by using an IC .In this section we didn’t face any problem and all results are accepted as we expected.*

*Finally, we interested when we do this experiment, since we learned how the different types of Adders, Subtractors, and Comparators are work.*