Communication Network Control Circuit

CSE 207 Project 2

Haibei Zhang, Section 2

Nov 6, 2002

All resources of this project are available at:

1. The Main Circuit (LogicWorks 4.1 circuit)
   http://www.haibei.net/cse207/Haibei_Proj2_Main.cct

2. The "Office Control Box" sub-circuit, with switches and probes for testing (LogicWorks 4.1 circuit)
   http://www.haibei.net/cse207/Haibei_Proj2_SubcctTEST.cct

3. The "Office Control Box" sub-circuit, ready to be converted into symbol (LogicWorks 4.1 circuit)
   http://www.haibei.net/cse207/Haibei_Proj2_SubcctFINAL.cct

4. The library which contains "Office Control Box" symbol
   (LogicWorks 4.1 library file)
   http://www.haibei.net/cse207/CSE207Lib.clf

5. This document (Adobe Acrobat)
   http://www.haibei.net/cse207/Haibei_Proj2_Report.pdf

My e-mail:
haibei@haibei.net

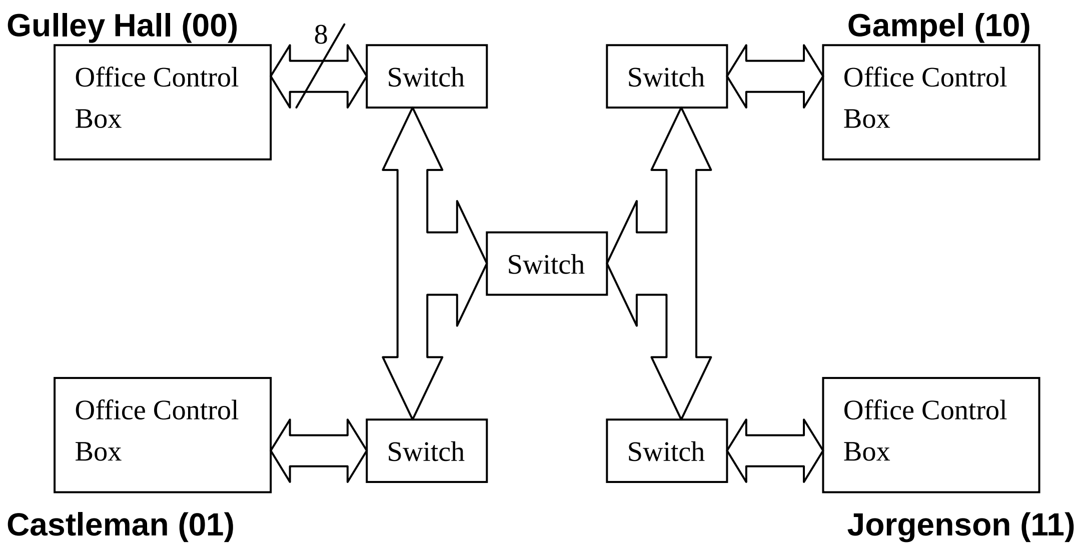
Objectives

Based on a variety of inputs asserted into the Office Control Box, the control circuits in the Office Control Box will generate signals to control the 5 Switches, which determine the status and direction of an 8-bit data transfer among the 4 offices.

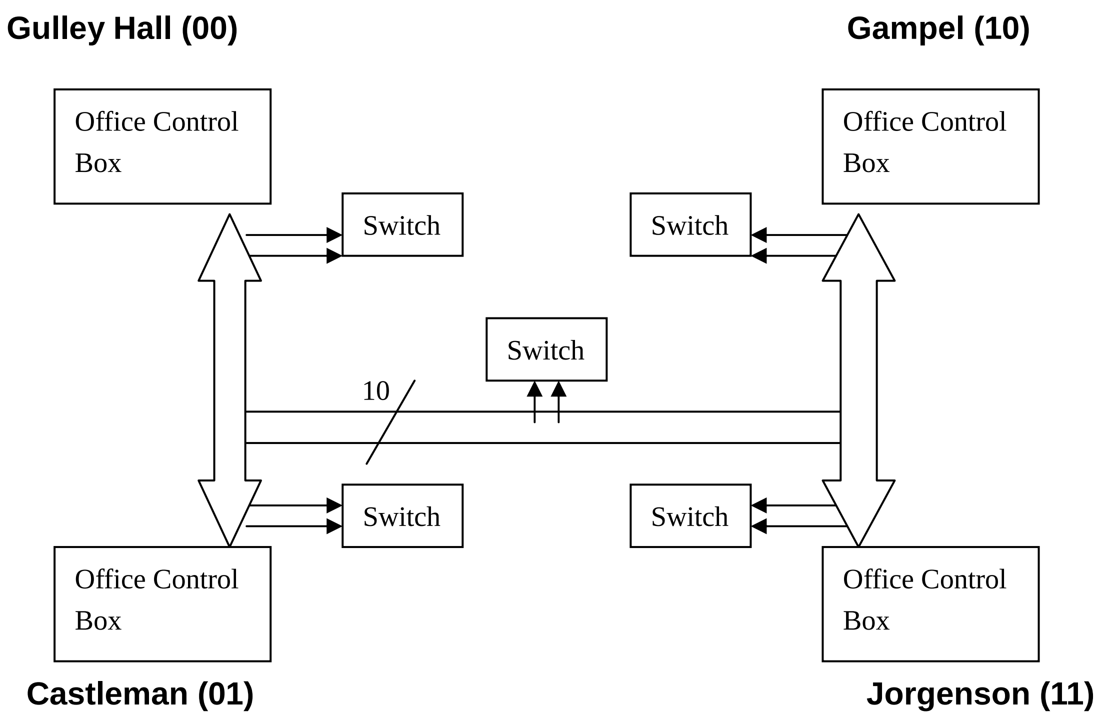
Design Procedures

1. Build the overall block diagram of the system (shown as follows).

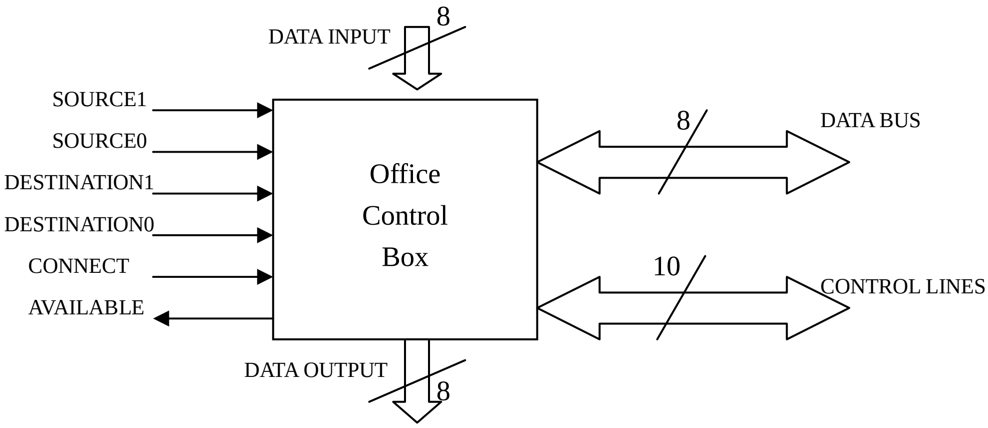
8-bit data bus:



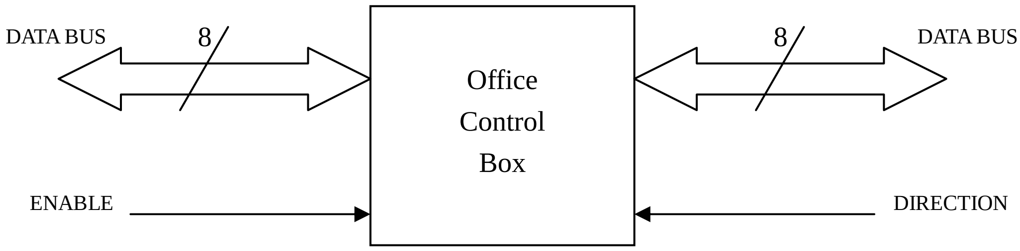
The 10 control lines:



The I/O of an Office Control Box



The I/O of an Switch



2. Based on above diagrams, find the relationship between inputs and outputs of the Control Box and the Switch.

3. Write the logical function of the relationship and simplify it if possible.

4. Build the circuit of that function.

5. Assemble all circuits in the Control Box or the Switch, convert it into a chip.

6. Use the new chip to build the whole system.

Functional Description

A. The Office Control Box:

1. Data outputs:
   The 8-bit data outputs are always connected with the 8-bit data bus. If the office is sending data, they output the same data as input. Otherwise they receive data from the data bus. In all cases their data are same as those on the bus.

2. Data inputs:
   The 8-bit data inputs can only be connected to data bus when the office is sending data. Otherwise it is disconnected. Therefore it is related to CONNECT (assuming the data bus is clear):

CONNECT	**DATA INPUTS**
1	Connected to data bus
0	Disconnected

   An 74x541 chip will fulfill this function. (As shown in the "Office Control" circuit)

   However, if the data bus is not clear (other office is using it), the CONNECT input is locked as inactive, hence the data inputs disconnected. This is to ensure that there will be no conflict on data bus. This will be discussed in "LOCKOUT FUNCTION (FOR EXTRA CREDIT)" section.

3. AVAILABLE (output)
   The AVAILABLE signal is based on the 3 (or 2, depends on path needed) ENABLE signals. The 3 (or 2) ENABLE will be selected from all 5 ENABLE signals from the control lines, among which 2 from ENABLE a, b, d, e and 1 is ENABLE c.

   - SRC1, SRC0 will select one ENABLE from a, b, d, e (a 4 input 1 bit mux will be used)

SRC1	SRC0	MUX's selection from EN_A, EN_B, EN_D, EN_E
0	0	EN_A
0	1	EN_B
1	0	EN_D
1	1	EN_E

   - DST1, DST0 will select one ENABLE from a, b, d, e (a 4 input 1 bit mux will be used) The MUX's selection is same as SRC1, SRC0 above.

   - EN_C is considered when data go across the C switch. Hence: $SRC1 \oplus DST1$ will determine whether ENABLE c will be considered. When $SRC1 \oplus DST1=1$, EN_C must be 1 (not used); When $SRC1 \oplus DST1=0$, we don't need C switch, so we don't care EN_C. The logical function is:
   $(SRC1 \oplus DST1)'+EN\_C$

   The product of above 3 outputs will be the AVAILABLE output

4. Setting of 5 ENABLE lines:
   If an office needs to send data, that Office Control Box will set 2 (or 3,

depends on whether the C switch is used) ENABLE signals to be active low.

- SRC1, SRC0 will set one ENABLE line from a, b, d, e (use 1/2 74_139 2-to-4 decoder)

SRC1	SRC0	EN_A	EN_B	EN_D	EN_E
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

(The ENABLE signals here are outputs of the decoder, not actual ENABLE in the control lines)

- DST1, DST0 will set one ENABLE line from a, b, d, e (use 1/2 74_139 2-to-4 decoder) The truth table is similar.

The actual ENABLE of a certain switch on the control lines is set to low, if either source or destination needs that switch. In order to combine the outputs from 2 decoders and convert Active High signals into Hi-Z, I use 4 open collector AND gates. Each gate takes 2 ENABLE from each decoder, and the output is the actual ENABLE signal on the control lines.

For example, the EN_A (from upper decoder) AND EN_A (from lower decoder) produces EN_A (actual signal on the control lines). If either source or destination requires A switch, one EN_A from the decoder is 0, hence the actual EN_A on the control lines is set 0 (active low).

- SRC1⊕DST1 will determine whether EN_C will be set:
EN_C= SRC1⊕DST1
Also, I add an open collector buffer to convert 1 into Hi-Z.

5. Setting of 5 DIRECTION lines:
If a switch is used to transfer data, and the direction is left-to-right, set that DIRECTION signal active low, otherwise don’t set it.

   If a source is on the left, or a destination is on the right, set the DIRECTION of the switch near that source (or destination) active low.

   - SRC1, SRC0 will set one DIRECTION line from a, b (use 1/2 74_139 2-to-4 decoder). Add an open collector buffer to decoder output to convert 1 into Hi-Z

   - DST1, DST0 will set one DIRECTION line from d, e (use 1/2 74_139 2-to-4 decoder). Add an open collector buffer to decoder output to convert 1 into Hi-Z

- The direction of C switch is left-to-right when SRC1=0 (a left source) and DST1=1 (a right destination)

DI_C=(SRC1’•DST1)’

Also add an open collector buffer to decoder output to convert 1 into Hi-Z.

6. CONNECT (input)
   CONNECT decides whether the Control Box can set control lines. CONNECT is asserted when the Office Control Box needs to send data. If CONNECT=0, all above circuits which affect the data bus or control lines are disabled (except Data Output and AVAILABLE, which only receives data from the data bus or control lines)

   - Data Input
   CONNECT’ is wired to the G1, G2 pin of the 74x541 chip. When CONNECT=0, the 74x541 disconnects all data inputs.

   - Setting of ENABLE
   When CONNECT=1, the circuit works as specified above. When CONNECT=0, all outputs should be Hi-Z, hence will not affect any control lines. To achieve this, CONNECT’ is wired to the G pin of both 1/2 74x139 chips. (when the chip is not enabled, all outputs are 1.) Besides, the EN_C is modified by adding an OR gate:
   EN_C=(SRC1⊕DST1)+CONNECT’
   so that when CONNECT=0, EN_C cannot be 0.

   - Setting of DIRECTION
   When CONNECT=1, the circuit works as specified above. When CONNECT=0, all outputs should be Hi-Z, hence will not affect any control lines. To achieve this, CONNECT’ is wired to the G pin of both 1/2 74x139 chips. (when the chip is not enabled, all outputs are 1.) Besides, the DI_C is modified by change the 2-input AND gate into an 3-input AND gate:

   DI_C=(SRC1’•DST1•CONNECT)’

   so that when CONNECT=0, DI_C cannot be 0.

B. **The Switch**
   Each switch is controlled by an ENABLE and a DIRECTION signal. The 74x245 transceiver does this job exactly.

Extra Design of “Lockout” Function (For Extra Credits)

Asserting a CONNECT signal (in order to send data) without notice of the availability of the system, is dangerous. The AVAILABLE output is now asserted as an input for CONNECT. However, since CONNECT and AVAILABLE are input for each other and changing each other, the circuit is unstable. I use a D-latch to fulfill the following logic requirement:

When CONNECT(input)=0, the ACTUAL CONNECT=0. (The office is not sending anything. No changes to CONNECT)

When CONNECT(input)=1, the AVAILABLE must also be 1 so that CONNECT can be successfully asserted (ACTUAL CONNECT=1). After it is successfully asserted, the AVAILABLE is changed to 0 (path occupied by this office). However, the CONNECT should "remember" the last AVAILABLE=1 and not change back to 0. The CONNECT can only set back to 0 by switching the CONNECT input manually.

When CONNECT(input)=1, AVAILABLE=0, the ACTUAL CONNECT will be 0, which means the CONNECT(input) is disabled. Again, this AVAILABILITY=0 will be remembered, which means the circuit cannot auto-detect the AVAILABILITY=1 and auto-assert the CONNECT. Once the AVAILABILITY becomes 1, the user has to manually switch the CONNECT input to 0, then back to 1, to assert it. This is designed on purpose in order to have better control of the whole system.

Examine the bottom-left of the "Sub-circuit of Office Control Box" circuit, where the D-latch is:

1. AVAILABLE is wired to D pin to be possibly remembered.
2. ACTUAL CONNECT is wired to Q pin, the output of D-latch
3. CONNECT' is wired to C pin so that when CONNECT=1, remember last D.
4. CONNECT is wired to R pin so that when CONNECT=0, set ACTUAL CONNECT(Q) =0.

By the way, the "auto-detecting AVAILABILITY" and "auto-asserting CONNECT" function can be implemented by wiring AVAILABLE to C pin, rather than CONNECT'.

<u>Testing</u>

Phase 1: Testing the "Office Control Box" sub-circuit
(using the "Office Control Box sub-circuit with switches and probes" circuit)

The observed results are as follows:

SRC	DST	EN_A	EN_B	EN_C	EN_D	EN_E	DI_A	DI_B	DI_C	DI_D	DI_E
00	01	0	0	1	1	1	0	1	1	1	1
00	10	0	1	0	0	1	0	1	0	0	1
00	11	0	1	0	1	0	0	1	0	1	0
01	00	0	0	1	1	1	1	0	1	1	1
01	10	1	0	0	0	1	1	0	0	0	1
01	11	1	0	0	1	0	1	0	0	1	0
10	00	0	1	0	0	1	1	1	1	1	1
10	01	1	0	0	0	1	1	1	1	1	1
10	11	1	1	1	0	0	1	1	1	1	0
11	00	0	1	0	1	0	1	1	1	1	1
11	01	1	0	0	1	0	1	1	1	1	1
11	10	1	1	1	0	0	1	1	1	0	1

The result matches the expected value perfectly. In order to obtain this result, two conditions must be met:

1. CONNECT=1. If CONNECT=0, all control signals are
2. Source cannot be same as destination. If they are same, the switch near that office will be enabled. For example, if SRC=DST=00, then EN_A=0. However, this is an abnormal situation.

Phase 2: Testing the whole system
(using the "Main Circuit")

Situation1: Normal situation (sending data from one place to another)
There are totally 12 possibilities in this situation(as shown in Phase 1):
From 00 to 01, 00 to 10, …… 11 to 10. For each case, set the Data Input at the source, then examine the Data Output at the destination to see if they match.
Result:
All data transferred successfully.

Situation 2: While sending data along a path, another office wants to use that path (or part of that path), too:
Cases tested:
1st user: 00 to 11 2nd user: 00 to 01
1st user: 00 to 11 2nd user: 10 to 01
1st user: 11 to 10 2nd user: 01 to 11
1st user: 10 to 00 2nd user: 11 to 01
Result:
The AVAILABLE of 2nd user is 0, the CONNECT of 2nd user is blocked out.

Situation 3: While sending data along a path, another office wants to use the unused path:
Case tested:

1st user: 00 to 01	2nd user: 11 to 10
1st user: 01 to 00	2nd user: 10 to 11
1st user: 10 to 11	2nd user: 00 to 01
1st user: 11 to 10	2nd user: 01 to 00

Result:
Data transferred successfully.

Discussion

This system still has following constraints:

1. In phase 2 testing, not all possibilities of situation 2 and 3 have been tested.
2. The source and destination can NOT be same. Otherwise the switch adjacent to that office is unnecessarily used. A future design of this system may adopt this logic: If SRC=DST, the Control Box shows an error but does not do anything else (if trying to assert CONNECT, block it), so that the Data Output can still receive data from the outside data bus.
3. Any two locations can NOT use a same SRC. Otherwise the system will generate unpredictable signals. To avoid this happening, use locked input device instead of binary switch for SRC1 and SRC0.

Conclusion

This communication network control circuit has shown its correctness and robustness (Lockout function!) in the testing phase. I am confident that, once deployed in UCONN campus, this system will perform as well as the PeopleSoft registration system, as specified by the requirement.