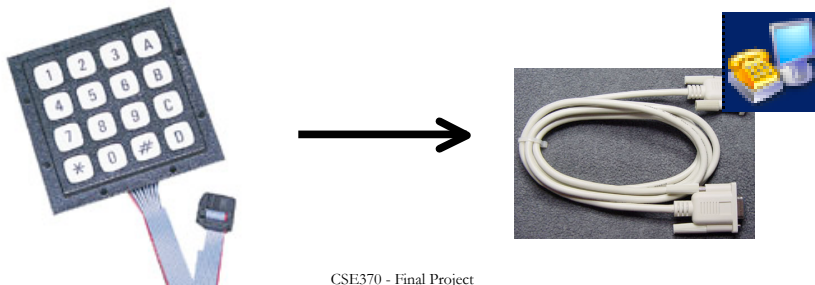


Final Project

- 16-key Keypad to RS-232 converter
- Solution will require 2 chips on the XLA5 protoboard
 - 2 – 22v10 PALs
- We'll provide everything but the core of the two PAL programs

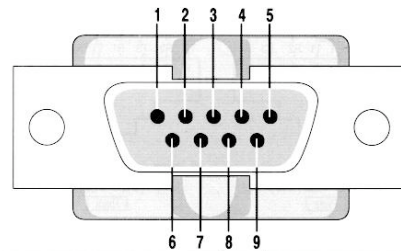


CSE370 - Final Project

1

Overview of RS232

- Very established serial line communication protocol
- Originally designed for teletypes and modems
 - Point-to-point, full-duplex
 - Variable baud (bit) rates
 - Cheap 9-wire connector connectors

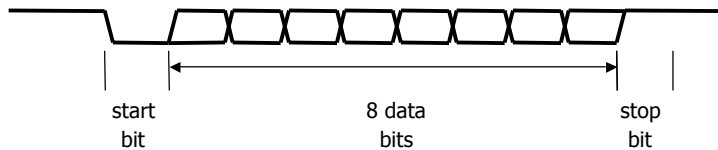


Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

CSE370 - Final Project

2

RS232 serial data format



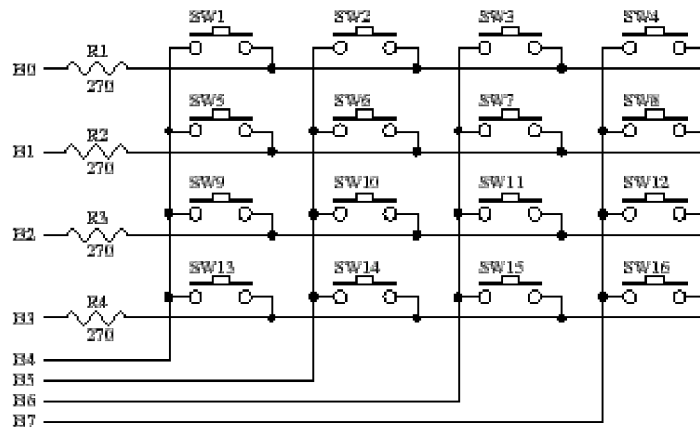
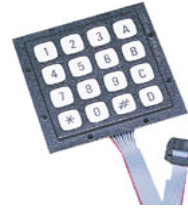
ASCII codes

- Each character has a unique code
- Some codes could be used to issue commands to the PC
 - E.g., clear, backspace, etc.

CHARACTER FONT DATA CODES
UPPER 4-BIT HEXADECIMAL

Lower 4-bit	Higher 4-bit	0	1	2	3	4	5	6	7	A	B	C	D	E	F	
0	xxxx0000	CS RAM (1)	0	1	2	3	4	5	6	7	A	B	C	D	E	F
1	xxxx0001	(2)	!	"	#	\$	%	&	'	()	*	+	,	-	
2	xxxx0010	(3)	.	/	0	1	2	3	4	5	6	7	8	9	:	
3	xxxx0011	(4)	;	<	=	>	?	@	A	B	C	D	E	F	G	
4	xxxx0100	(5)	H	I	J	K	L	M	N	O	P	Q	R	S	T	
5	xxxx0101	(6)	U	V	W	X	Y	Z	[\]	^	_	~	DEL	
6	xxxx0110	(7)	SP	!	"	#	\$	%	&	'	()	*	+	,	
7	xxxx0111	(8)	.	/	0	1	2	3	4	5	6	7	8	9	:	
8	xxxx1000	(9)	;	<	=	>	?	@	A	B	C	D	E	F	G	
9	xxxx1001	(A)	H	I	J	K	L	M	N	O	P	Q	R	S	T	
A	xxxx1010	(B)	U	V	W	X	Y	Z	[\]	^	_	~	DEL	
B	xxxx1011	(C)	SP	!	"	#	\$	%	&	'	()	*	+	,	
C	xxxx1100	(D)	.	/	0	1	2	3	4	5	6	7	8	9	:	
D	xxxx1101	(E)	;	<	=	>	?	@	A	B	C	D	E	F	G	
E	xxxx1110	(F)	H	I	J	K	L	M	N	O	P	Q	R	S	T	
F	xxxx1111	(F)	U	V	W	X	Y	Z	[\]	^	_	~	DEL	

Matrix Keyboard

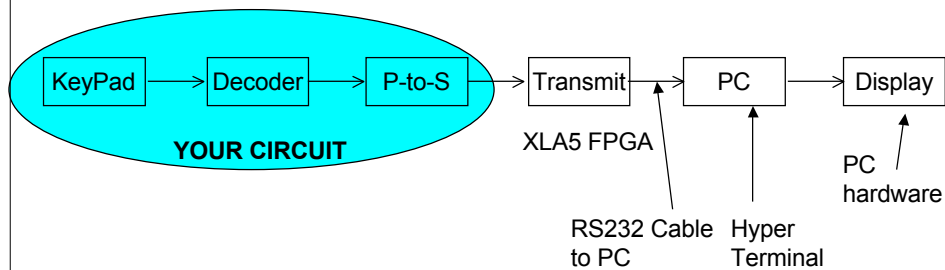


CSE370 - Final Project

5

Block diagram

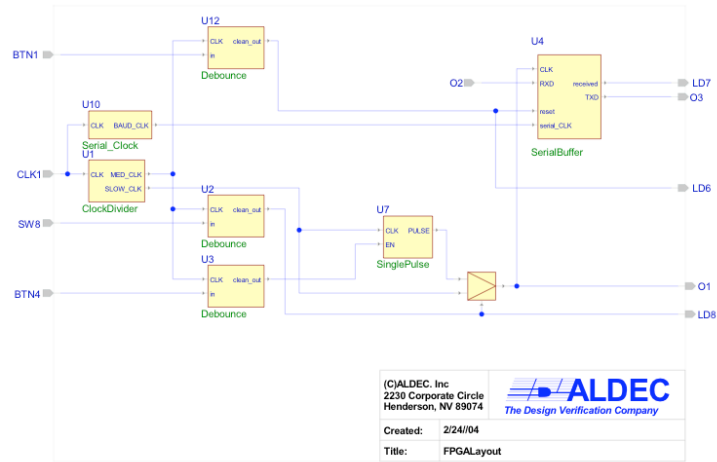
- Major components
 - Keypad
 - Keypad Decoder
 - **Parallel-to-serial converter**
 - PC display



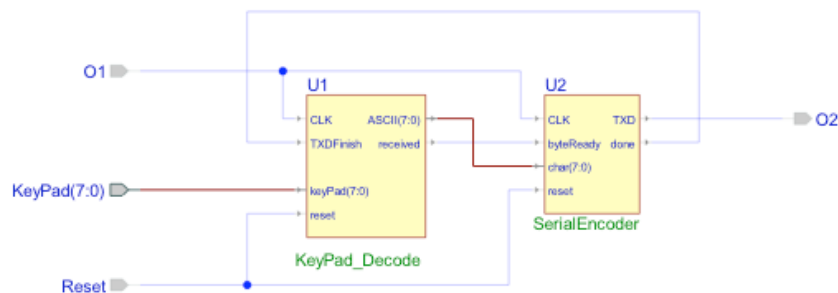
CSE370 - Final Project

6

XLA5 FPGA Model



Simulation model



Keypad_Decode – Decode the pressed keys

```
module Keypad_Decode ( CLK, TXDFinish ,ASCII ,keyPad ,received, reset);
    input TXDFinish ;
    wire TXDFinish ;
    input [7:0] keyPad ;
    wire [7:0] keyPad ;
    input CLK;
    wire CLK;
    input reset;
    wire reset;

    output received ;
    reg received ;
    output [7:0] ASCII ;
    reg [7:0] ASCII ;

    always@(posedge CLK)
        begin
            ;
        end

endmodule
```

SerialEncoder – Sends the data to the PC

```
module SerialEncoder ( CLK, TXD ,byteReady ,done ,char, reset);

    input byteReady ;
    wire byteReady ;
    input [7:0] char ;
    wire [7:0] char ;
    input reset;
    wire reset;
    input CLK;
    wire CLK;

    output TXD ;
    reg TXD ;
    output done ;
    reg done ;

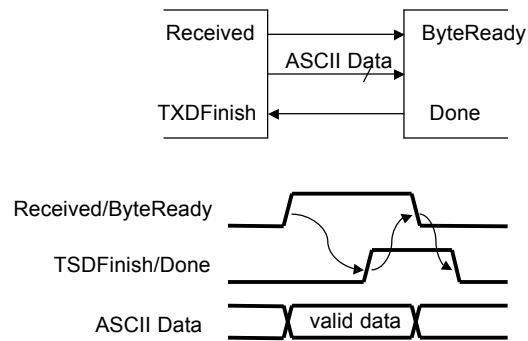
    always @(posedge CLK)
        begin

        end

endmodule
```

Four-cycle handshake between modules

- Don't let one get ahead of the other



Purpose of the project

- Learn how to build a realistic system
- Read data sheets
- Communicating state machines
- Deal with existing code/components