Critical Design Review:

By:Smart Guitar Group
Alex Paige
Larry Zhao
Henry Tang
Jeff Hanna

Table of Contents

1. Overview
2. Processor
   - Pin Configuration
   - Interface Specifications
   - Memory Mapping
   - Timing
3. Power Design
   - Analog Power
   - Digital Power
   - Guitar Pick-up Power
4. Components/Schematic
   - Bluetooth, SDRAM, SD card, Roland Pickup...
5. Bill of Materials
6. Software Design

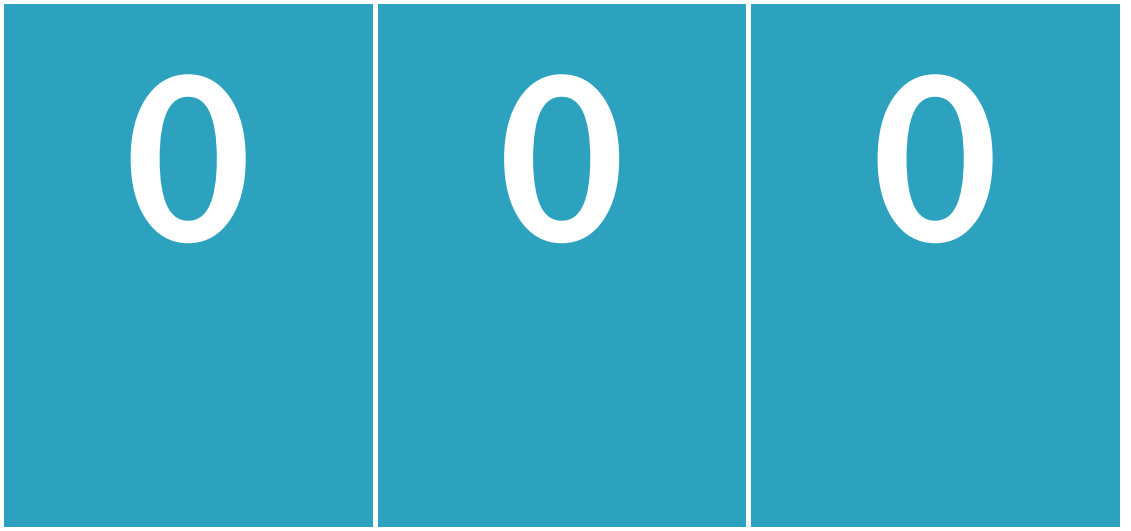
Mode Overview

- Tuning
  - While in Tuning mode, the note being played is displayed on the LCD screen. Three LED's indicate whether the input is flat, sharp, or in tune.
- Write to Memory
  - While in this mode, the data produced by the FFT is stored in memory to be sent via bluetooth at another time.
- Live
  - This is the main operating mode. While in this mode FFT data is sent via bluetooth to a computer which will write the music score in live time.

Setting Time Signature and BPM

- The three digits can be incremented individually using the up buttons under them. If the input is unchanged for three seconds it will begin to blink, indicating that it is about to be set.
- After three blinks the Time signature has been set and the BPM is set using the same procedure. When the BPM blinks, it counts off the time signature at the current BMP rate and the FFT is started.

User Interface



Tuning:

Mode: ^ 0 0 0

Power: 0

Power Design (Vdd & Gnd)

Power: 3.3V

-Digital: 15,60,71,89,112,125,146,165,181,198,26,86,174,38

-Analog: 20,24

Gnd:

-Digital: 33,63,77,93,114,133,148,169,189,200,32,84,172

-Analog: 22

Memory Mapping

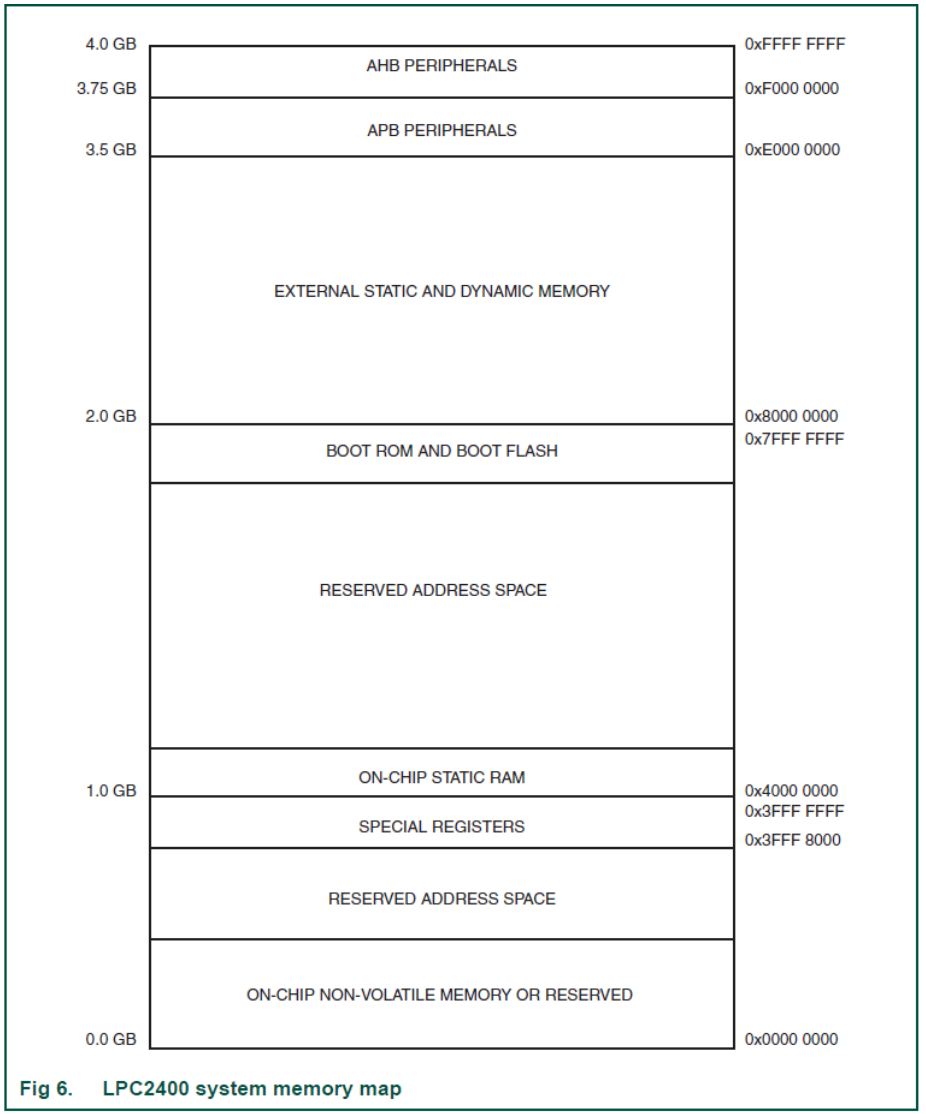


Fig 6. LPC2400 system memory map

OnBoard Flash:

- To store program, 512k
- Enough for our purpose

QPNano Statemachine

- ARM C lib
- FFT code
- Sys Init & Peripheral Code

OnBoard RAM:

- 64K
- Enough for running program and in-place 1024 point FFT

Interface Specification:

UART0: Debug/On-Board Flash Programming Port
UART2: Bluetooth
Interrupt based: register: **U0IIR - 0xE000 C008 U0IIR - 0xE000 C008**

From Interrupt source(bit 3:1) to figure out which status register to read next

011 1 - Receive Line Status (RLS).
010 2a - Receive Data Available (RDA).
110 2b - Character Time-out Indicator (CTI).
001 3 - THRE Interrupt

Timing Concern:

Can the system run fast enough?
Our sampling rate **44khz = 0.023ms**

Our target play rate is set at 5 times/sec per string, which equal to 30 times FFT calculation. **1/30=33ms**

Sampling takes about 11clock cycle per input channel, at 20mhz, it takes about 3microsec for all input channels.

According to following table, we can calculate 512 sample FFT at about **6ms**

Thus, ADC channels will have to use timed interrupt to take samples at sampling frequency, and use the rest of the time to calculate FFT.

Timing Concern (Chart)

Size of FFT N	40 MHz ARM7		40 MHz ARM7M	
	Time for one FFT	FFTs per second	Time for one FFT	FFTs per second
16	0.028ms	36,000	0.026ms	38,000
32	0.080ms	12,500	0.069ms	14,500
64	0.20ms	4,700	0.17ms	5,700
128	0.52ms	1,900	0.41ms	2,400
256	1.24ms	800	1.00ms	1,040
512	2.9ms	340	2.2ms	450
1024	6.6ms	150	4.9ms	200
2048	15ms	67	11ms	89
4096	33ms	30	25ms	41
8192	73ms	14	53ms	19
16384	160ms	6.3	115ms	8.7

Table 1: Optimized timings

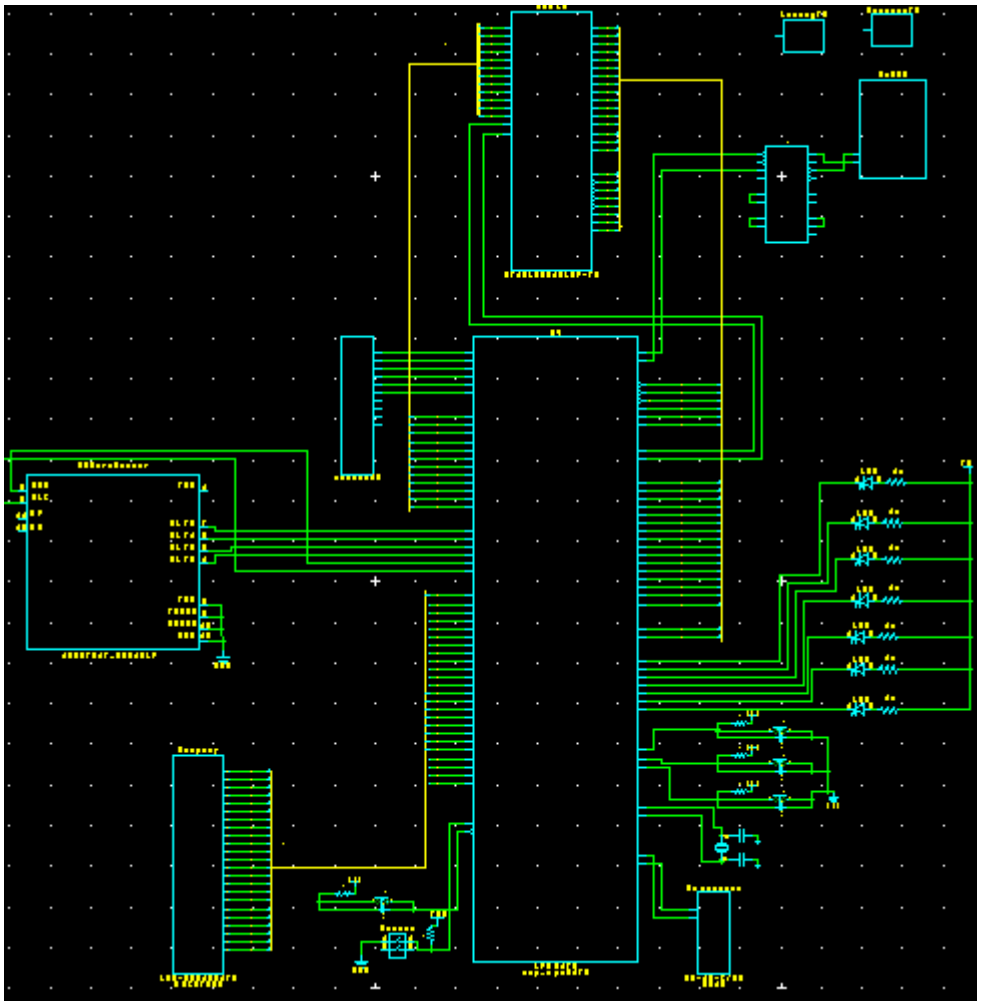
Other software concern:

Note Time Stamp:

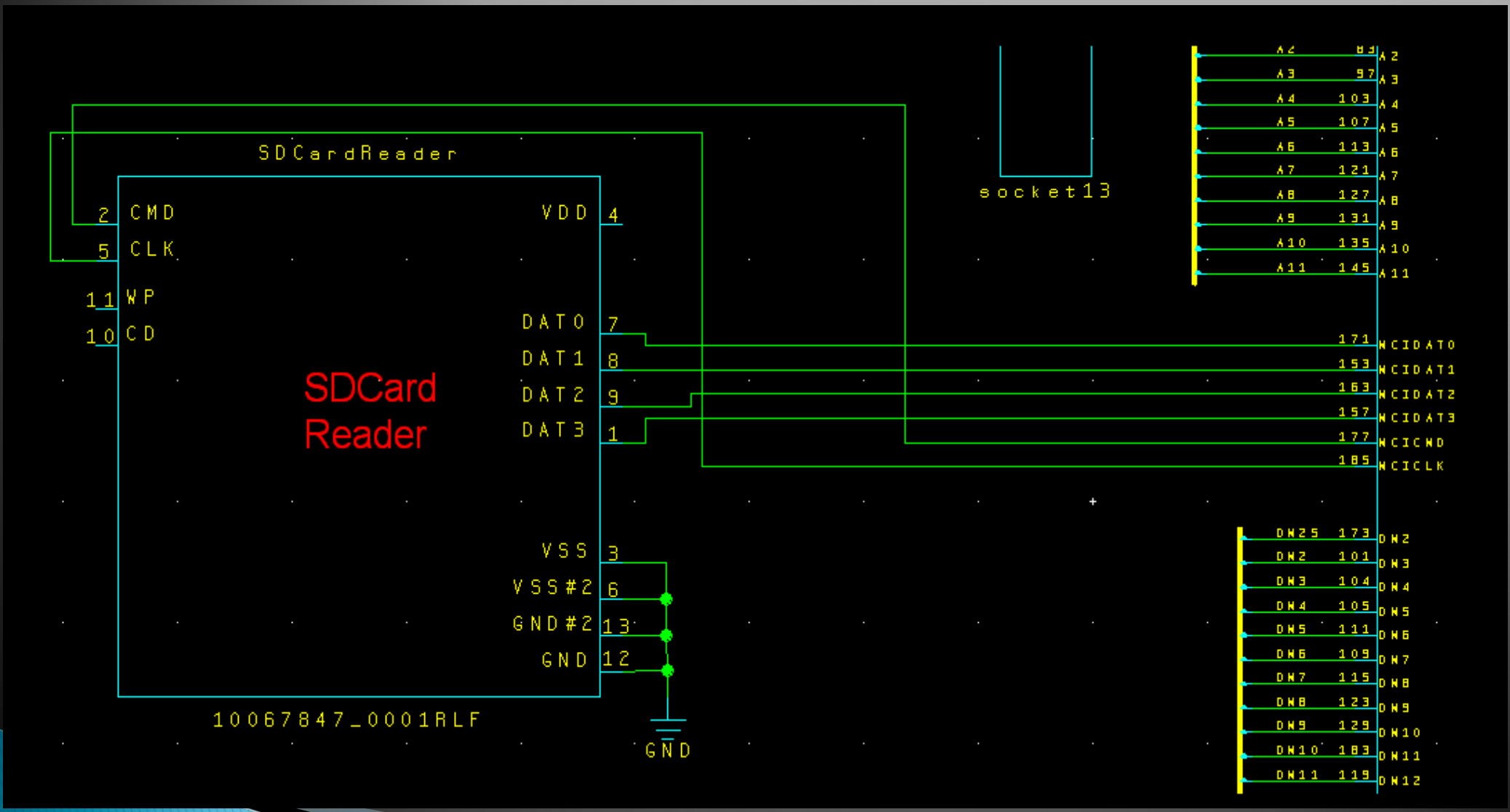
A threshold will be in place to detect string play, and the note will be time stamped. Even though the note will not be determined until later time due to FFT delay.

Schematic

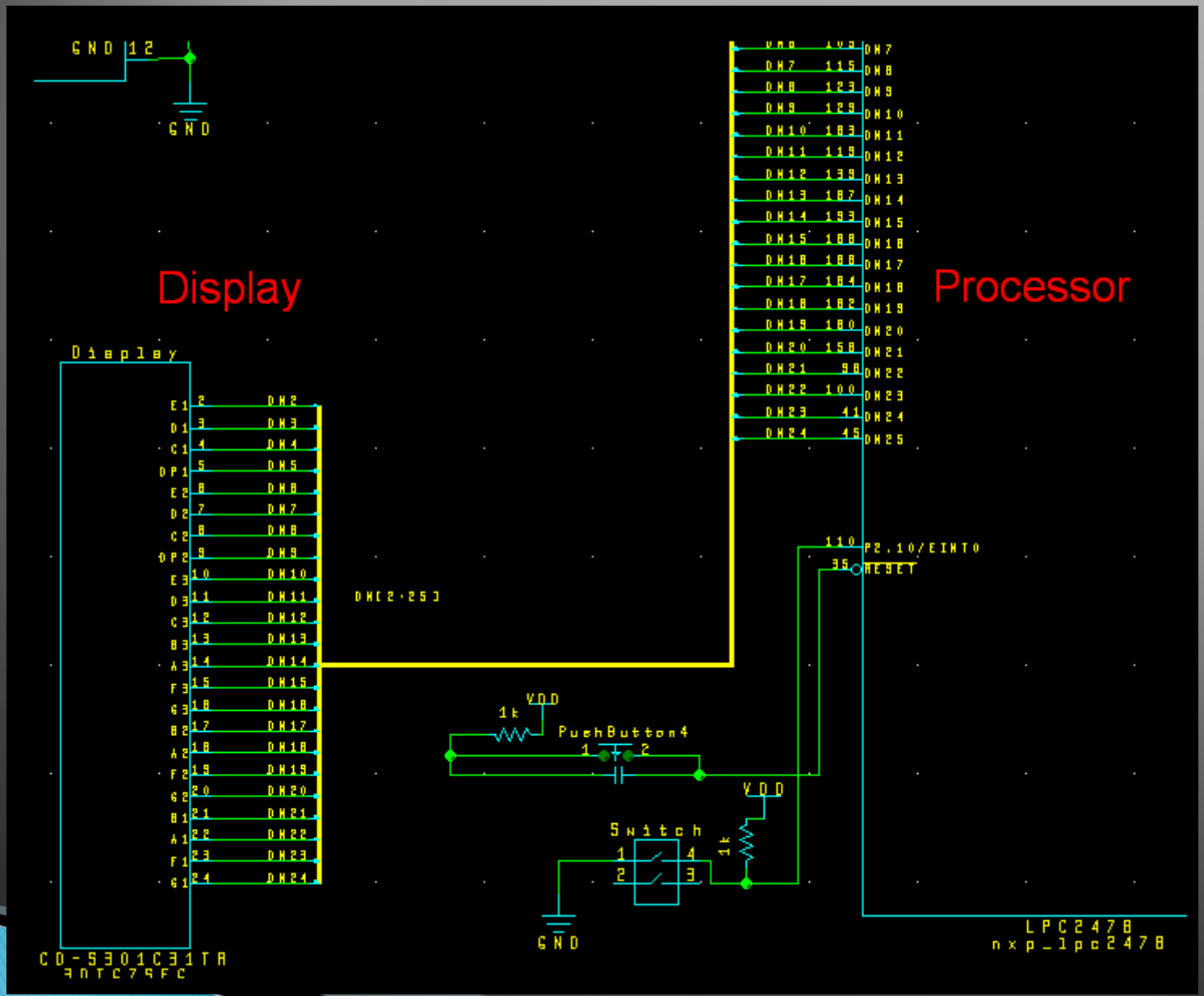
- SDCard Reader
- 24-pin Display
- Rs232 (w/Level Shifter)
- 7 LED's
- Bluetooth
- SDRam
- Pushbutton
- 13-pin Socket



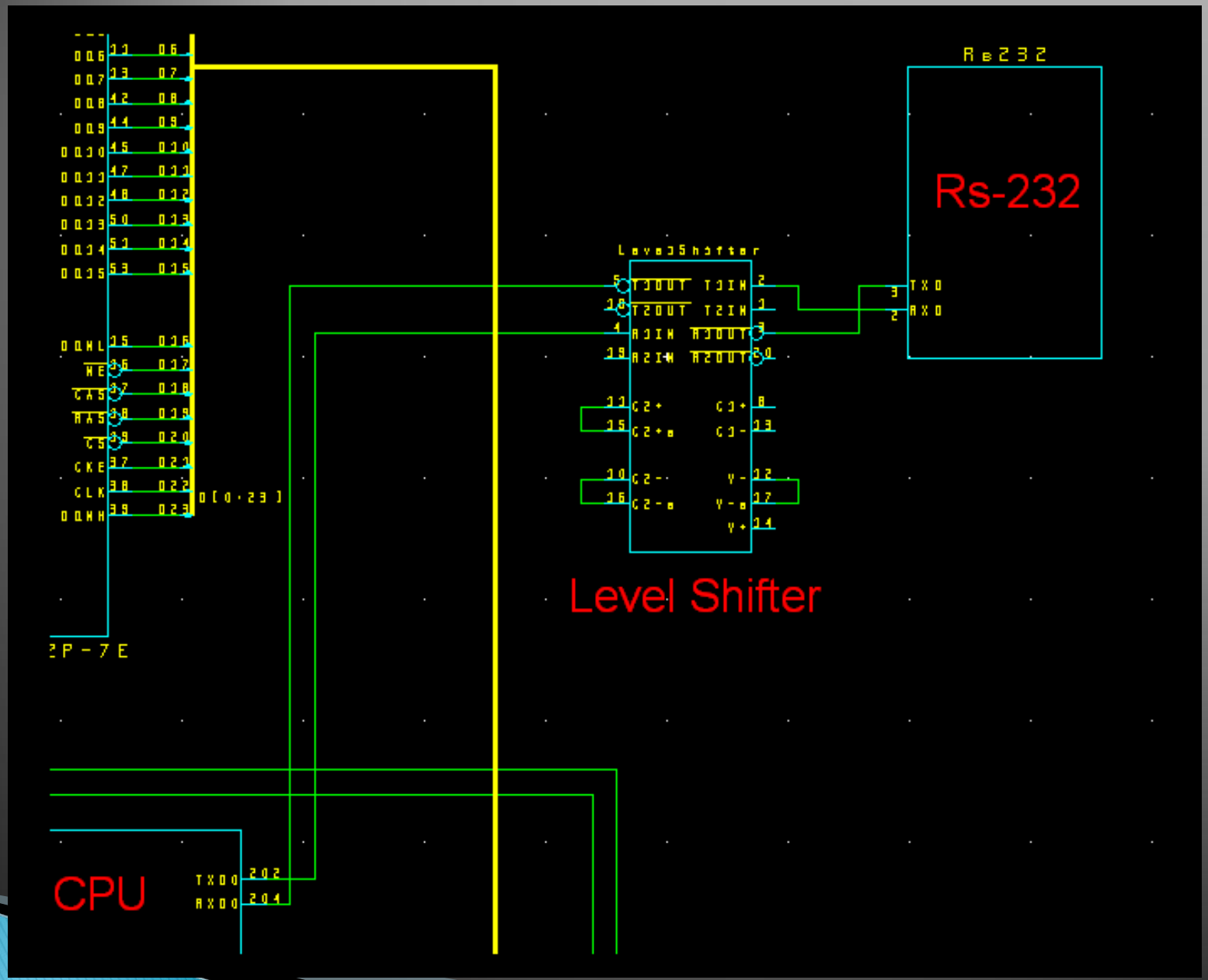
SDCard Reader



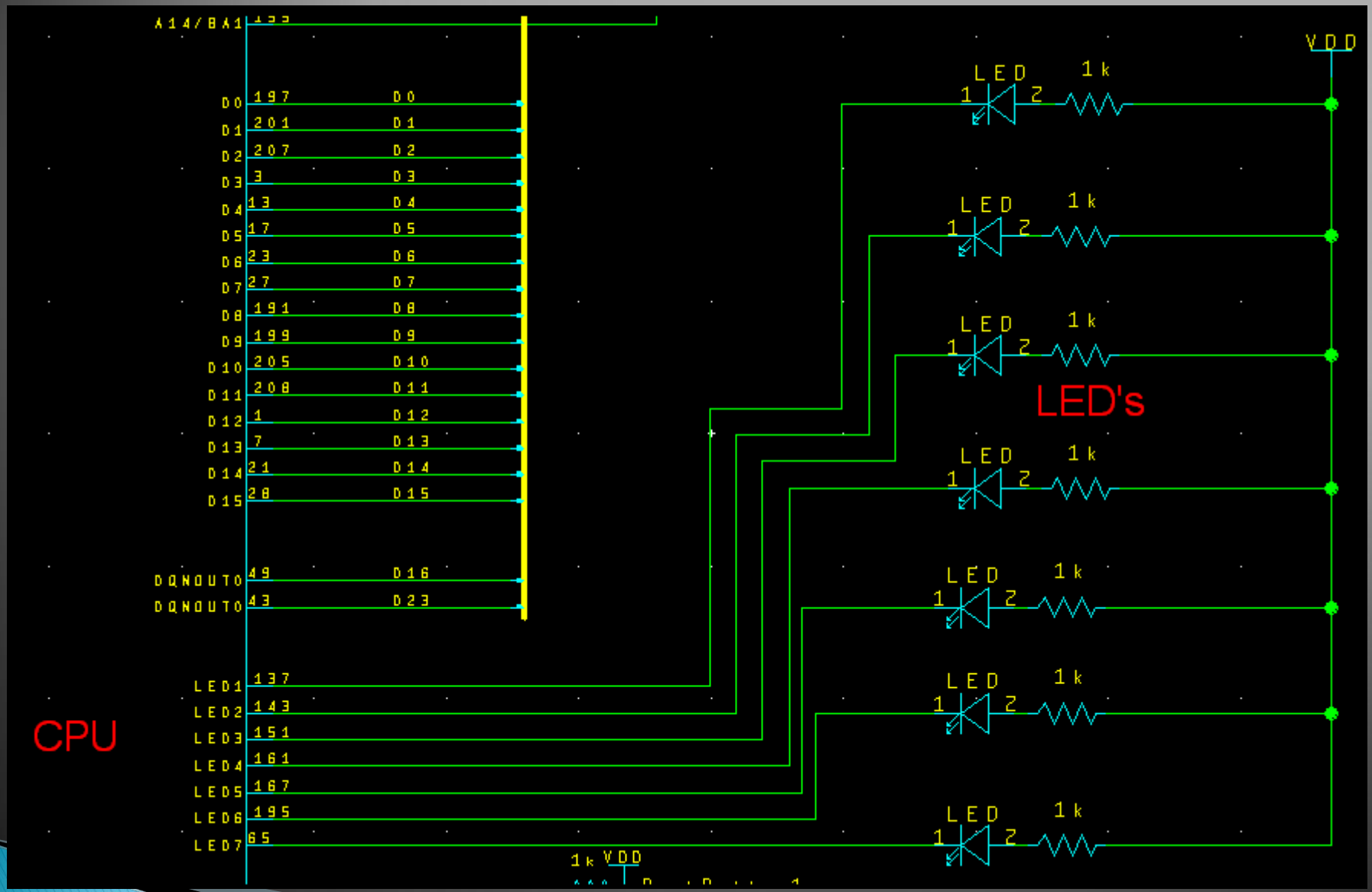
24-pin Display



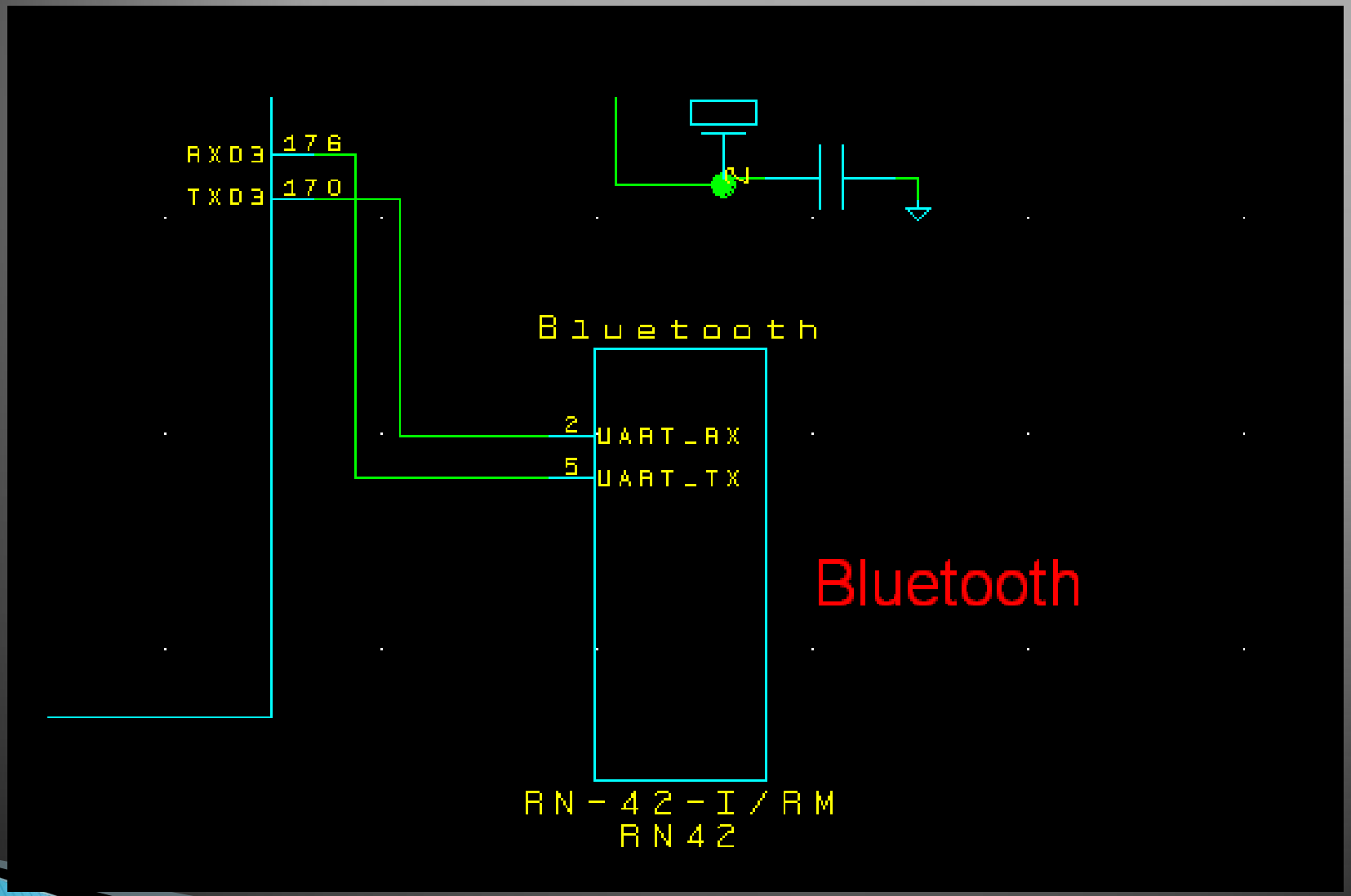
RS232 and Level Shifter



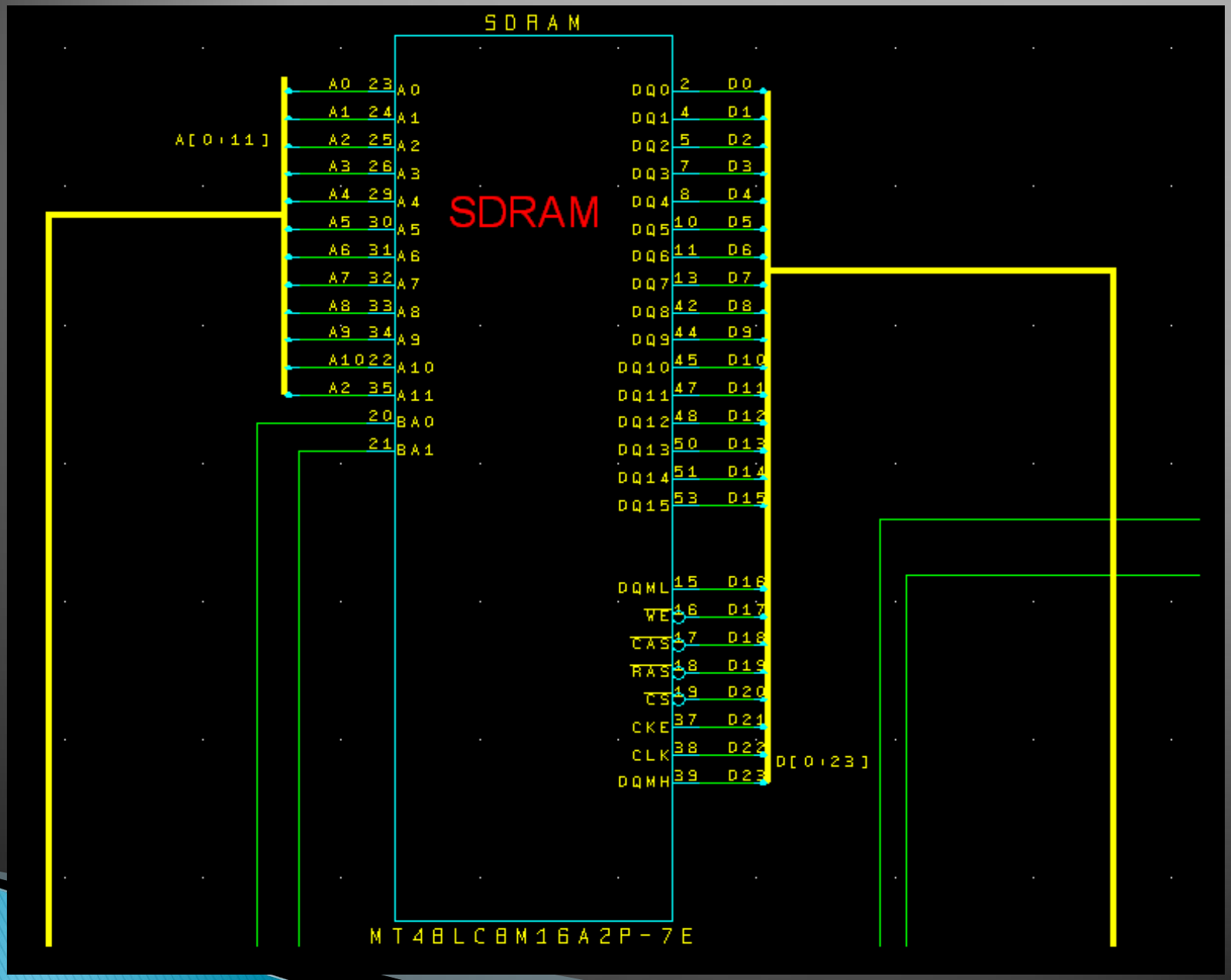
LED's


A14/BA1
D0 197 D0
D1 201 D1
D2 207 D2
D3 3 D3
D4 13 D4
D5 17 D5
D6 23 D6
D7 27 D7
D8 191 D8
D9 199 D9
D10 205 D10
D11 208 D11
D12 1 D12
D13 7 D13
D14 21 D14
D15 28 D15
DQNOUT0 49 D16
DQNOUT0 43 D23
LED1 137
LED2 143
LED3 151
LED4 161
LED5 167
LED6 195
LED7 65
CPU
1k VDD
VDD
LED 1k
1 2
LED's

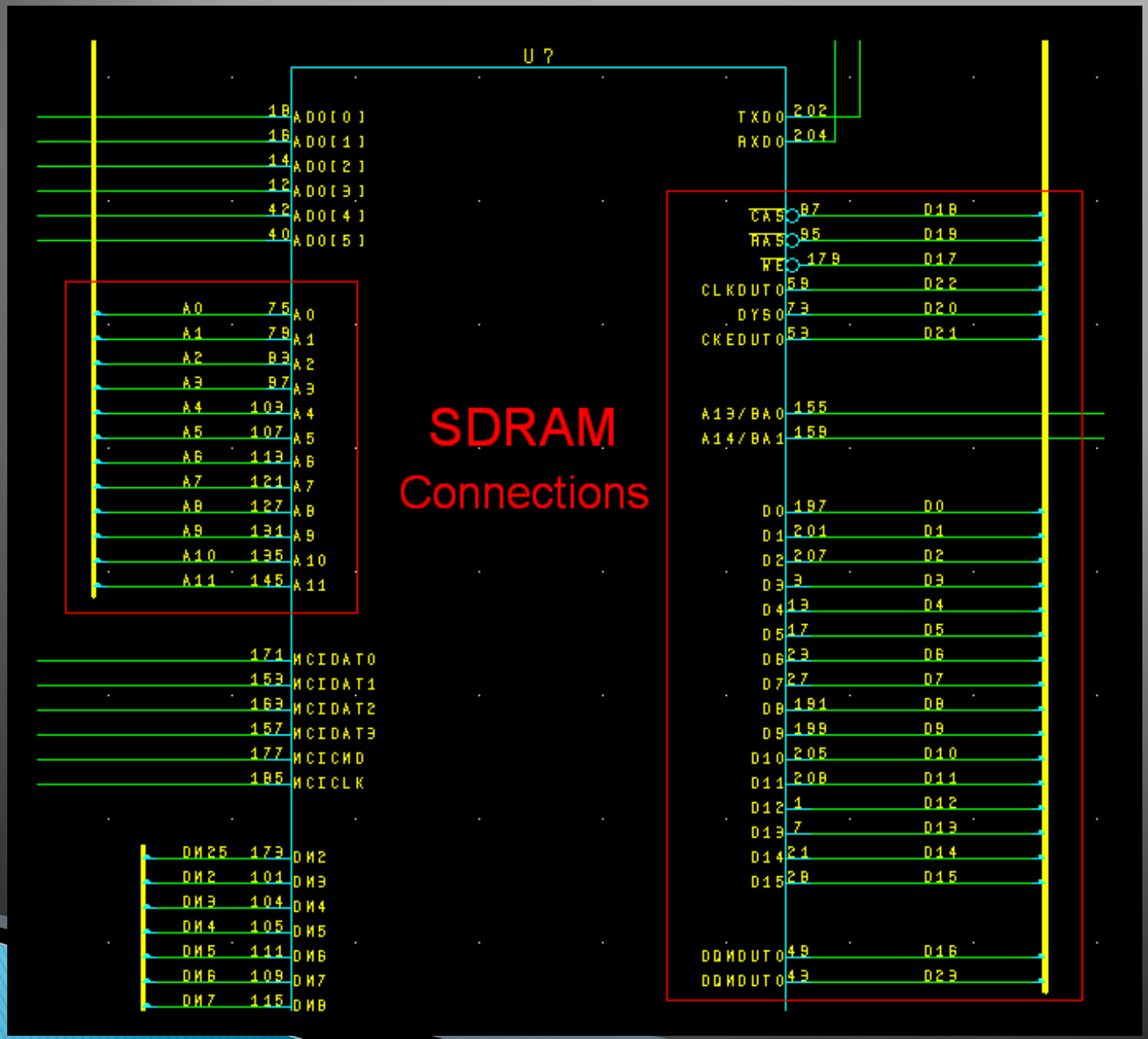
Bluetooth


RXD3
176
TXD3
170
Bluetooth
2
UART_RX
5
UART_TX
Bluetooth
RN-42-I/RM
RN42

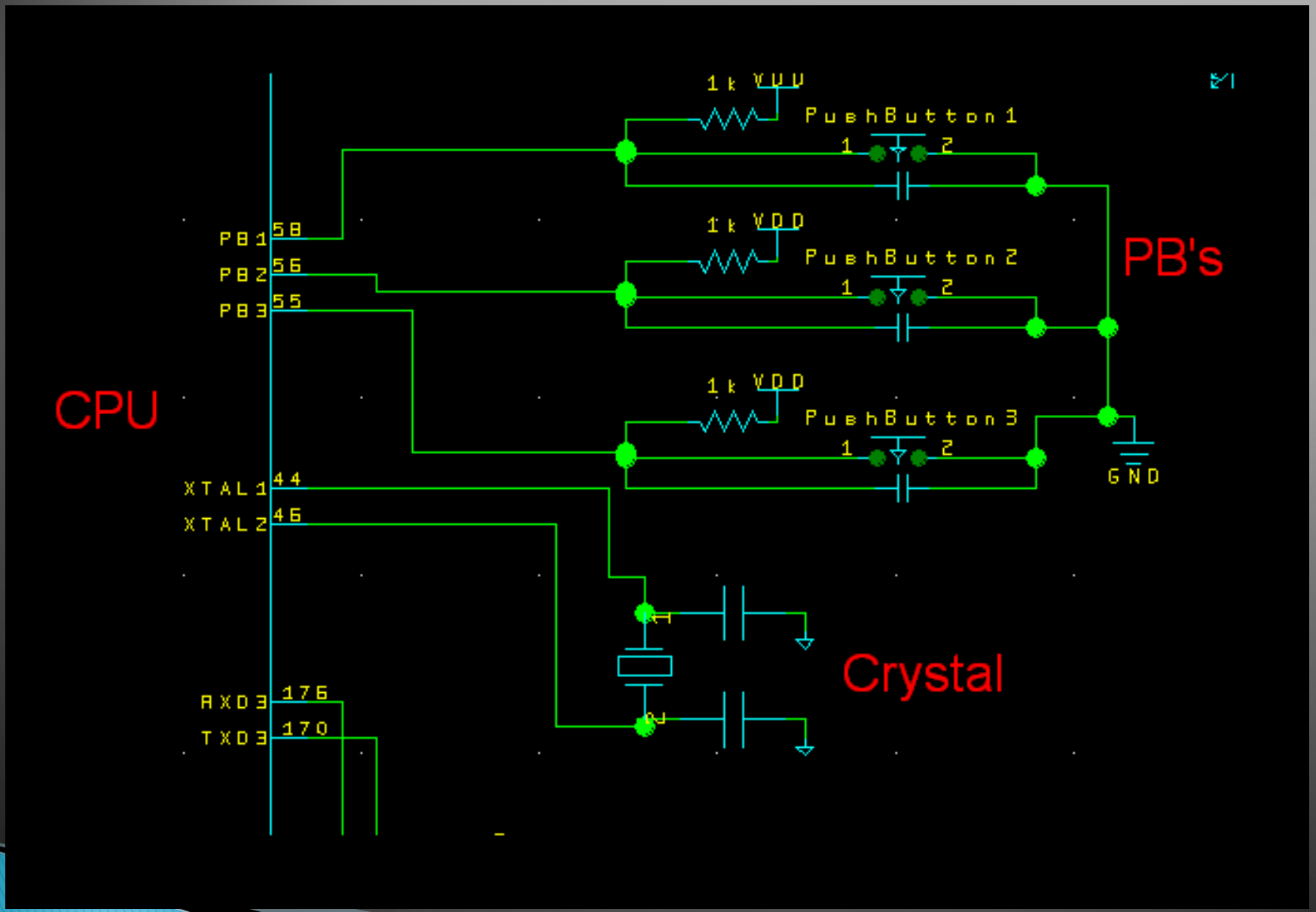
SDRAM (Part 1)


SDRAM
A[0..11]
A0 23 A0
A1 24 A1
A2 25 A2
A3 26 A3
A4 29 A4
A5 30 A5
A6 31 A6
A7 32 A7
A8 33 A8
A9 34 A9
A10 22 A10
A2 35 A11
20 BA0
21 BA1
SDRAM
DQ0 2 D0
DQ1 4 D1
DQ2 5 D2
DQ3 7 D3
DQ4 8 D4
DQ5 10 D5
DQ6 11 D6
DQ7 13 D7
DQ8 42 D8
DQ9 44 D9
DQ10 45 D10
DQ11 47 D11
DQ12 48 D12
DQ13 50 D13
DQ14 51 D14
DQ15 53 D15
DQML 15 D16
WE 16 D17
CAS 17 D18
RAS 18 D19
CS 19 D20
CKE 37 D21
CLK 38 D22
DQMH 39 D23
D[0..23]
MT48LC8M16A2P-7E

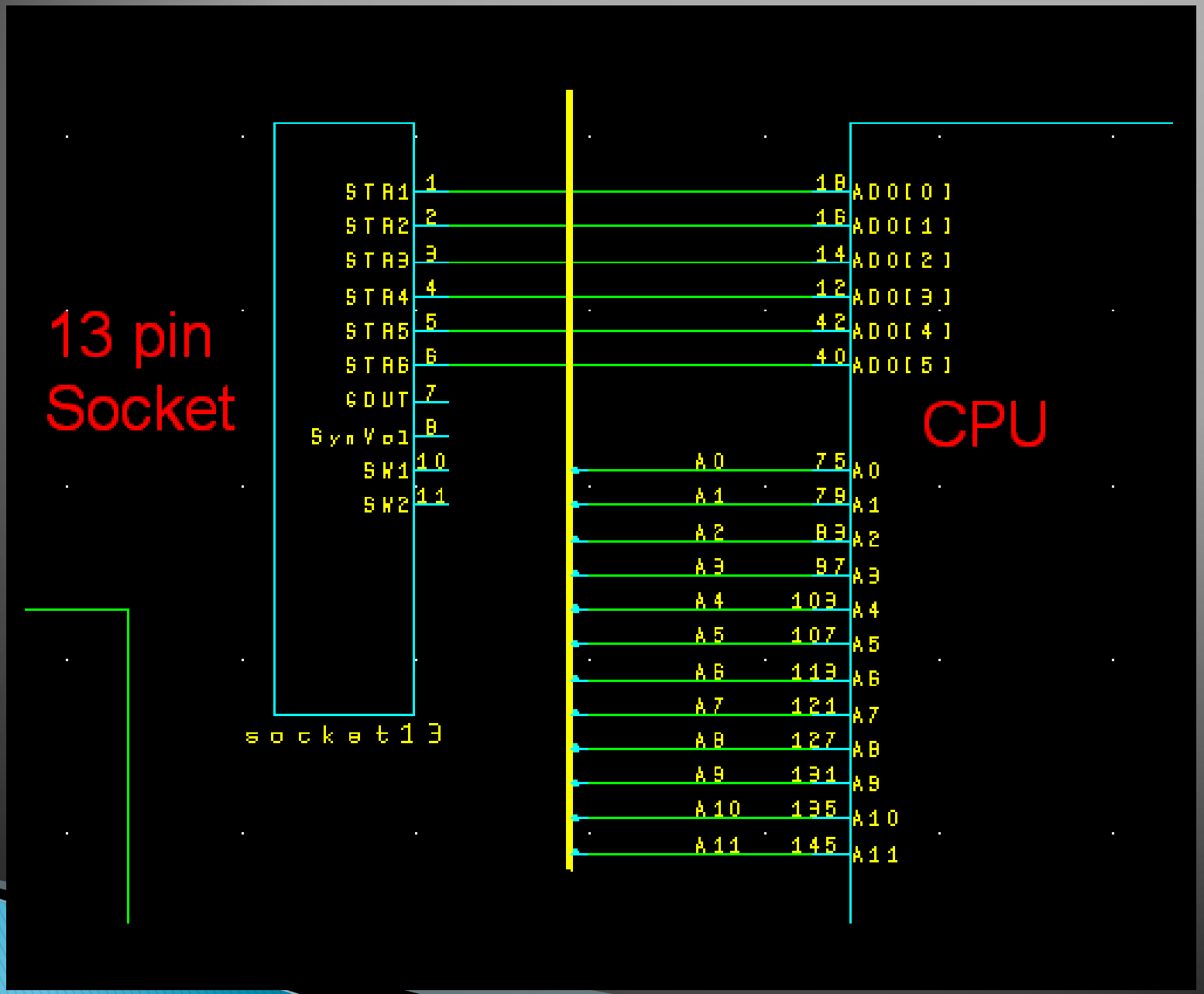
SDRAM (Part 2)


U ?
SDRAM
Connections
ADO[0]
ADO[1]
ADO[2]
ADO[3]
ADO[4]
ADO[5]
TXDO
RXDO
CAS
RAS
WE
CLKOUTO
DYSO
CKEOUTO
A13/BA0
A14/BA1
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
MCIDAT0
MCIDAT1
MCIDAT2
MCIDAT3
MCICMD
MCICLK
DQMOUTO
DQMOUTO

PushButton and Crystal



13-pin socket



Software Architecture

1. QPNano state machine
   Open source, ARM compilable.
   Three State: Tune, Record, Transmit

2. Optimized FFT
   Fixed point FFT, and uses cos/sin lookup table.
   Algorithm and code can be found online at:
   http://www.jjj.de/fft/fftpage.html

   No special math library needed.