

Nexys4 Audio Player

Andrew Reiff
Daniel Vega
Daniel French

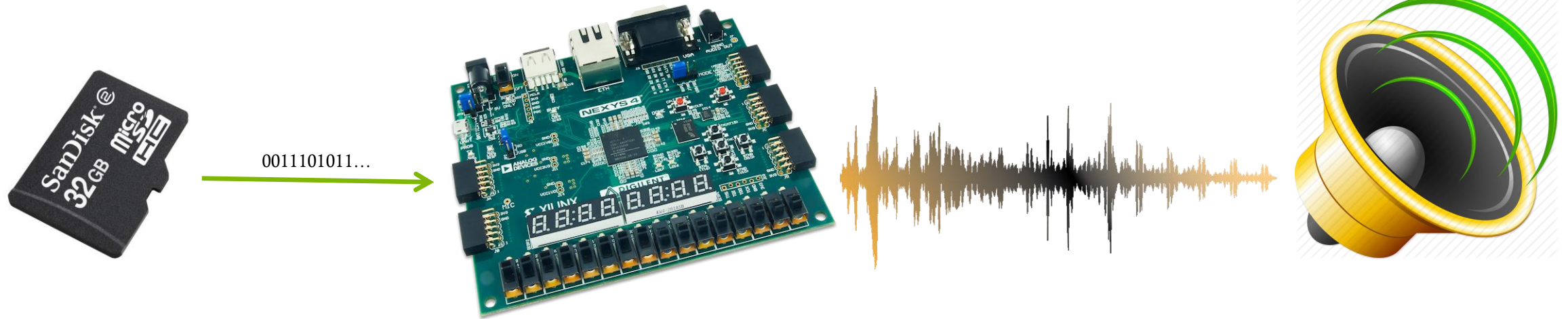
CSE 378
Winter 2016



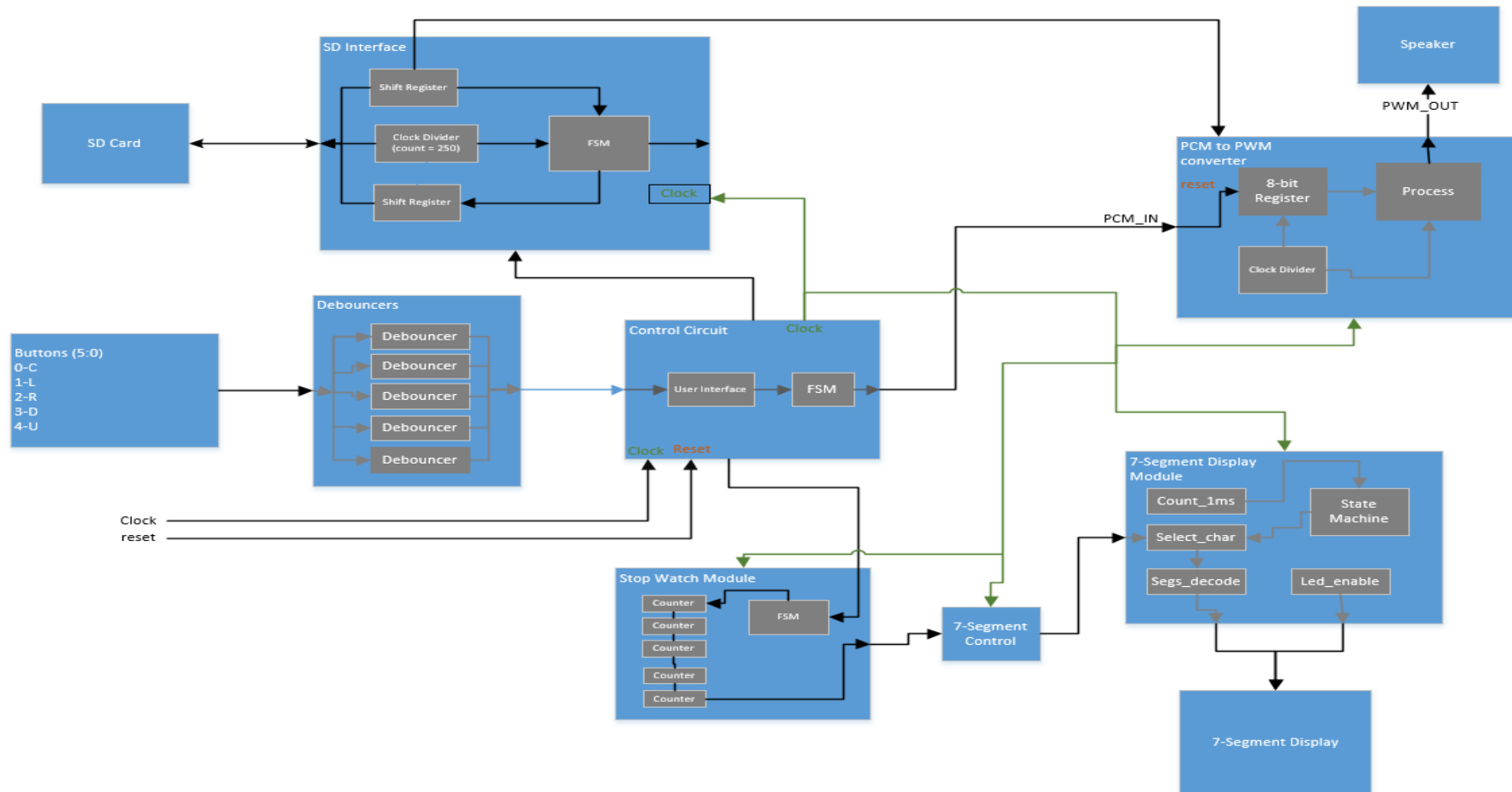
Contents

- ◆ Top Level Design
- ◆ Interfacing the SD card with Nexys4
- ◆ PCM vs. PWM signals
- ◆ Seven Segment Clock (Timer)
- ◆ Further Design Implementations

Design



Top Level Design Architecture



Reading From SD Card

- ◆ Clock
 - ◆ 100-400MHz
- ◆ Communication Method
 - ◆ Secure Digital (SD)
 - ◆ Serial Peripheral Interface (SPI)
- ◆ Format
 - ◆ FAT16 vs FAT32
- ◆ Pinouts
 - ◆ Different for each method
 - ◆ Initialization will fail if no pullup's

Pin	Name	Function (SD Mode)	Function (SPI Mode)
1	DAT3/CS	Data Line 3	Chip Select/Slave Select (SS)
2	CMD/DI	Command Line	Master Out Slave In (MOSI)
3	VSS1	Ground	Ground
4	VDD	Supply Voltage	Supply Voltage
5	CLK	Clock	Clock (SCK)
6	VSS2	Ground	Ground
7	DAT0/DO	Data Line 0	Master In Slave Out (MISO)
8	DAT1/IRQ	Data Line 1	Unused or IRQ
9	DAT2/NC	Data Line 2	Unused

Table 1: SD Card Pin Assignments [2].

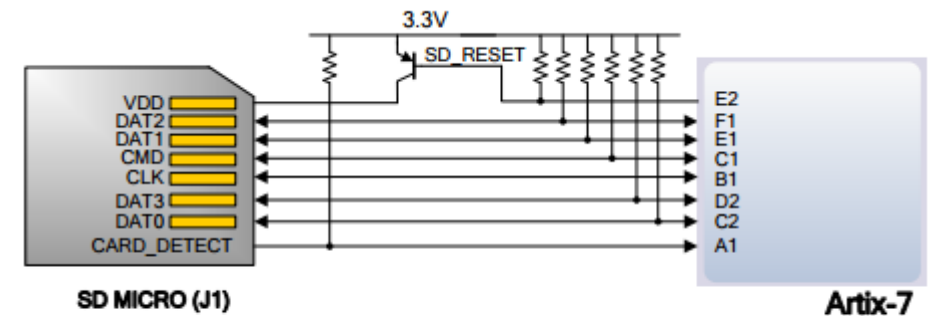


Figure 21. Artix-7 microSD card connector interface (PIC24 connections not shown).

SD Protocol

- ◆ Operation:
 - ◆ Single data line (DAT0) or four data lines (DAT0-DAT3).
 - ◆ Command is transferred serially on the CMD line.
 - ◆ Response is transferred serially on the CMD line.
 - ◆ Data is transferred in Blocks
 - ◆ Block are always followed by CRC (confirmation) bits
- ◆ Process:
 - ◆ Command (48 bits) to SD card
 - ◆ Command (CMD17)= “001000” for read
 - ◆ Argument = Data Address
 - ◆ 8 clock cycles (SD card)
 - ◆ Response (48 bits) from SD card
 - ◆ Receive .wav data packet

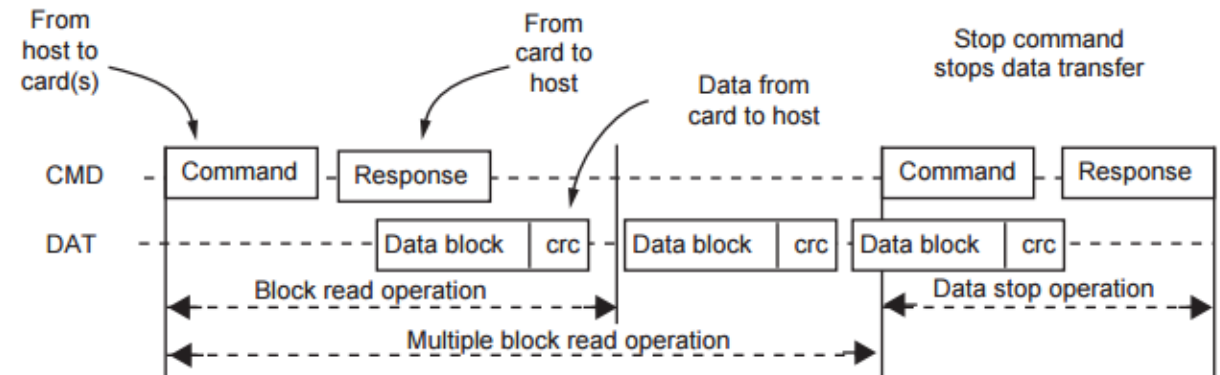
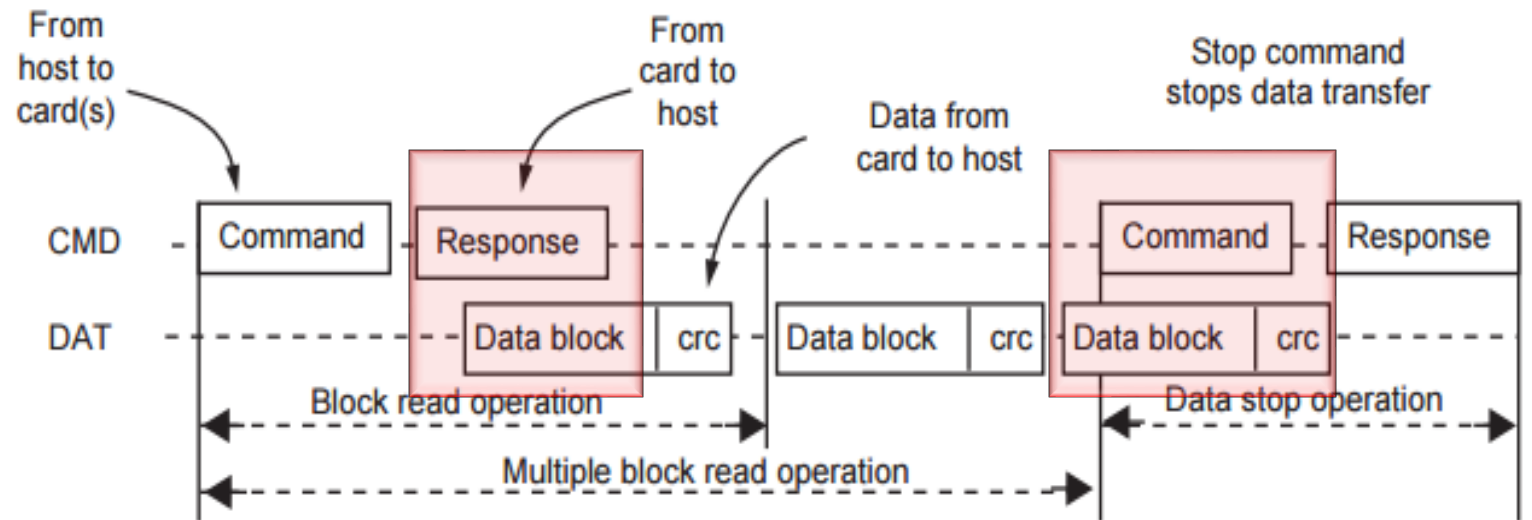


Figure 1: Multiple block read operations

0	1	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	host	command	argument	CRC7 ¹	end bit

Table 1: Command and Response format

SD Protocol Timing



SPI Protocol

Initialization

- ◆ DI and SD = 1
- ◆ Wait minimum 90 clock cycles
- ◆ CMD0 = 0x00000000
- ◆ Sent CMD1 to check status
- ◆ Response change from Idle to ready (0x01 → 0x00)

Process

- ◆ Send command signal (48 bits)
- ◆ Wait 8 clock cycles for SD card to process
- ◆ SD card sends 48-bit response
- ◆ Command Response Time (N_{CR})
 - ◆ 0 to 8 bytes
- ◆ SD card then sends data stream

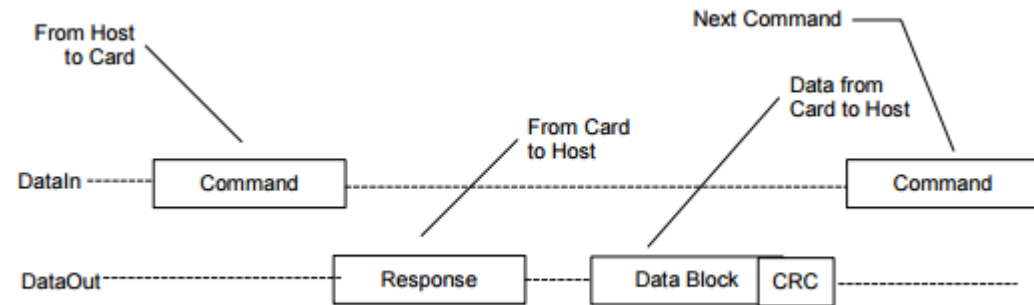


Figure 1. Single block read operation

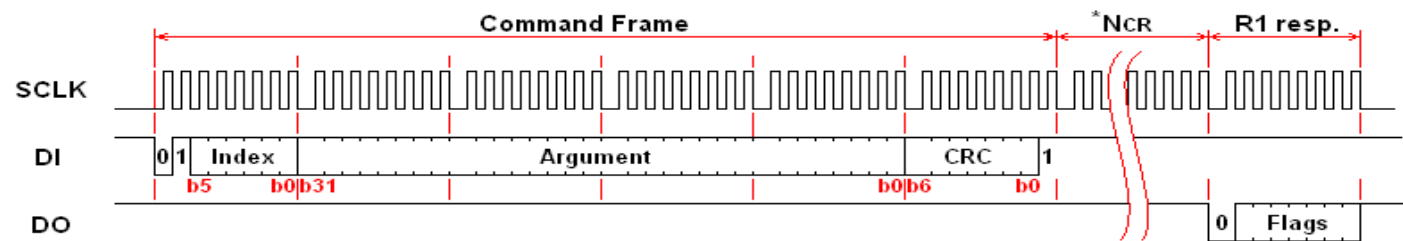


Figure 2. Timing of single block read operation

SD (Secure Digital Card) Uncertainties

- ◆ FAT32 SD card formatting
 - ◆ Where exactly is the data located within the SD card file directory?
- ◆ Timing of the .wav file message
- ◆ Was the entire length of the .wav file being received?

SD Format

- File Allocation Table (FAT)
 - Disk divided into clusters
 - First 512 bytes is boot sector
 - Cannot be changed
 - Stores information about disk
 - 4 to 64 sectors per cluster
 - Clusters determine where a file is located
- FAT16
 - Cards 128MB to 2GB
- FAT32
 - Cards 2GB to 32GB

HxD - [BLACKBERRY (I:)]

File Edit Search View Analysis Extras Window ?

16 ANSI hex Sector 473 of 3853824

BLACKBERRY (I:)

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F		
0003B200	42	4C	41	43	4B	42	45	52	52	59	20	08	00	13	05	6D	BLACKBERRYm	
0003B210	70	3F	70	3F	00	00	26	2E	8C	48	00	00	00	00	00	00	p?p?...&.GH.....	
0003B220	41	2E	00	5F	00	2E	00	54	00	72	00	0F	00	7F	61	00	A...T.r...a.	
0003B230	73	00	68	00	65	00	73	00	00	00	00	FF	FF	FF	FF	FF	s.h.e.s....YVYV	
0003B240	7E	31	20	20	20	20	20	20	54	52	41	22	00	00	00	29	~1 TRA"...	
0003B250	8C	48	8C	48	00	00	00	29	8C	48	06	00	00	10	00	00	GHGH...)GH.....	
0003B260	4D	41	52	49	4E	20	20	20	57	41	56	20	10	89	B8	19	MARIO WAV %. .	
0003B270	8C	48	8C	48	00	00	B8	19	8C	48	5B	00	3A	EA	00	00	GHGH...)GH[:ë..	
0003B280	41	2E	00	54	00	72	00	61	00	73	00	0F	00	25	68	00	A..T.r.a.s...\$h.	
0003B290	65	00	73	00	00	FF	FF	FF	FF	00	00	FF	FF	FF	FF	FF	e.s...YVYV..YVYV	
0003B2A0	54	52	41	53	48	45	7E	31	20	20	20	12	00	00	00	29	TRASHE~1	
0003B2B0	8C	48	8C	48	00	00	00	29	8C	48	04	00	00	00	00	00	GHGH...)GH.....	
0003B2C0	42	30	00	30	00	00	00	FF	FF	FF	FF	0F	00	21	FF	FF	B0.O...YVYV.!YV	
0003B2D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	YVYVYVYVYV..YVYV	
0003B2E0	01	2E	00	53	00	70	00	6F	00	74	00	0F	00	21	6C	00	...S.p.o.t...!l.	
0003B2F0	69	00	67	00	68	00	74	00	2D	00	00	00	56	00	31	00	i.g.h.t.-...V.l.	
0003B300	53	50	4F	54	4C	49	7E	31	20	20	20	12	00	01	00	29	SPOTLI~1	
0003B310	8C	48	8C	48	00	00	00	29	8C	48	07	00	00	00	00	00	GHGH...)GH.....	
0003B320	41	2E	00	66	00	73	00	65	00	76	00	0F	00	DA	65	00	A..f.s.e.v...Ûe.	
0003B330	6E	00	74	00	73	00	64	00	00	00	00	FF	FF	FF	FF	FF	n.t.s.d....YVYV	
0003B340	46	53	45	56	45	4E	7E	31	20	20	20	12	00	0C	00	29	FSEVEN~1	
0003B350	8C	48	8C	48	00	00	00	29	8C	48	10	00	00	00	00	00	GHGH...)GH.....	
0003B360	E5	41	55	4C	54	20	20	20	20	20	10	00	64	3B	8A		âAULT ..d;Š	
0003B370	90	46	8C	48	00	00	BB	12	8C	48	60	02	00	00	00	00	.FGEH...»..GH.....	
0003B380	42	20	00	49	00	00	6E	00	66	00	6F	00	0F	00	72	72	00	B .I.n.f.o...rr.
0003B390	6D	00	61	00	74	00	69	00	6F	00	00	00	6E	00	00	00	m.a.t.i.o...n...	
0003B3A0	01	53	00	79	00	73	00	74	00	65	00	0F	00	72	6D	00	.S.y.s.t.e...rm.	
0003B3B0	20	00	56	00	6F	00	6C	00	75	00	00	00	6D	00	65	00	.V.o.l.u...m.e.	
0003B3C0	53	59	53	54	45	4D	7E	31	20	20	20	16	00	4A	03	14	SYSTEM~1 ..J..	
0003B3D0	8C	48	8C	48	00	00	04	14	8C	48	02	00	00	00	00	00	GHGH...GH.....	
0003B3E0	E5	2E	00	5F	00	4D	00	41	00	52	00	0F	00	D6	49	00	â...M.A.R...ŌI.	
0003B3F0	4F	00	2E	00	77	00	61	00	76	00	00	00	00	00	FF	FF	O...w.a.v....YV	
0003B400	E5	4D	41	52	49	7E	31	20	57	41	56	22	00	89	26	2E	âMARI~1 WAV"%. .	
0003B410	8C	48	8C	48	00	00	26	2E	8C	48	58	00	00	10	00	00	GHGH...&.GHX.....	
0003B420	E5	61	00	6E	00	61	00	6C	00	6F	00	0F	00	D0	67	00	âa.n.a.l.o...Dg.	
0003B430	5F	00	77	00	72	00	69	00	74	00	00	00	65	00	5F	00	.w.r.i.t...e..	
0003B440	45	4E	41	4C	4F	47	7E	31	50	4E	47	20	00	80	96	1A	âANALOG~1PNG .€~.	

Offset: 3B260 Block: 3B260-3B264 Length: 5 Readonly Overwrite

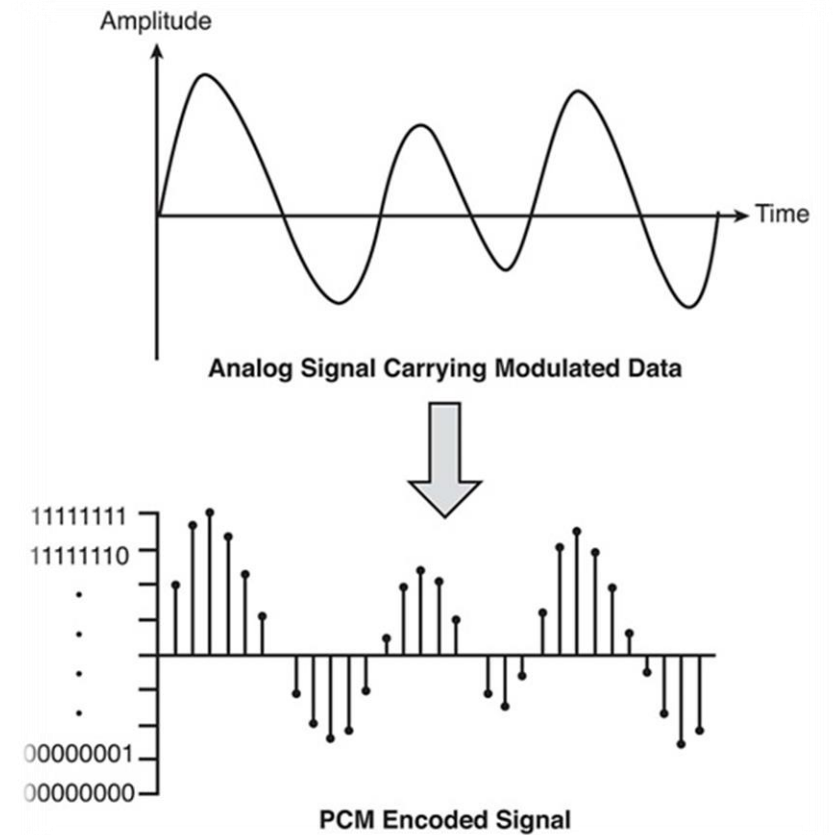
FAT: What We Know

Based on the Boot Sector Information

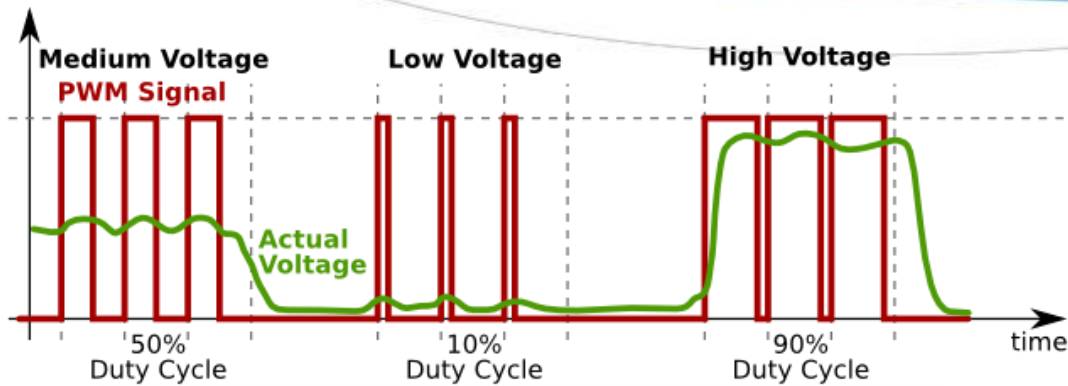
- ◆ 512 bytes per sector
- ◆ 64 sectors per cluster
- ◆ 1 reserved sector
- ◆ 512 Root Directory Entries
- ◆ 3854329 Sectors in File System
- ◆ 2 FAT copies
- ◆ 236 Sectors per FAT
- ◆ FAT #1
 - ◆ Offset: 512 - 121343
- ◆ FAT #2
 - ◆ Offset: 121344 - 242175
- ◆ Total Cluster Size: 32,768 bytes

PCM (Pulse Code Modulation)

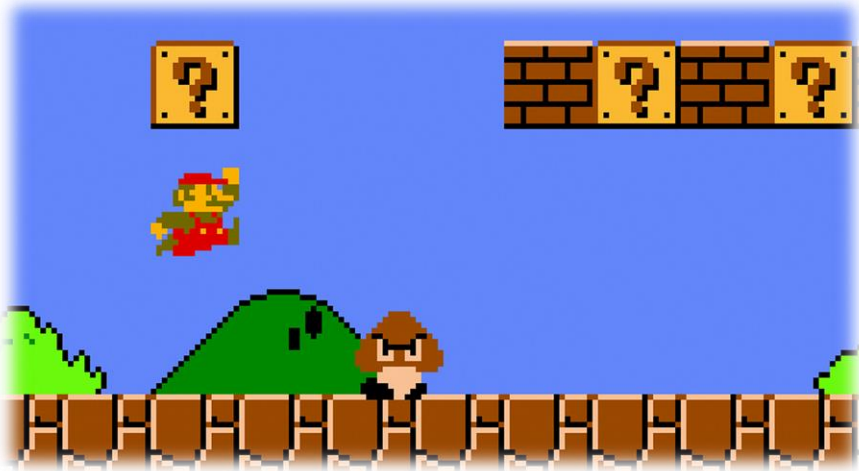
- Method used to digitally represent a sampled Analog Signal
- Amplitude of the signal is sampled at regular intervals
 - Standard form of digital audio in computers, compact discs, etc.
- Each Amplitude or step is “quantized” to the nearest value in a set of digital steps



PWM (Pulse Width Modulation)

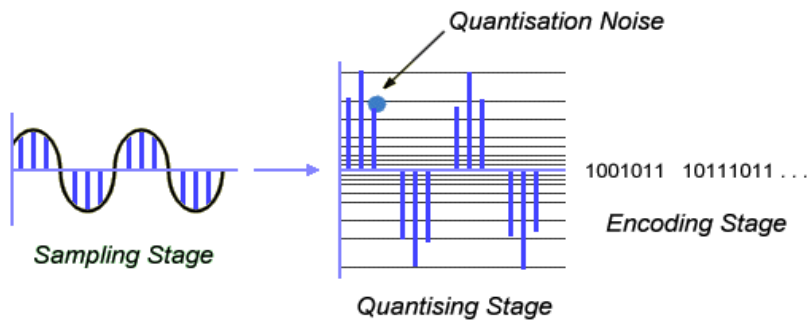


- Modulation technique used to encode a message in a pulsing signal
- Used to control the power supplied to electrical devices
- PWM has been used to play back a crude version of a PCM signal
 - Speakers driven by two voltage levels, 0V and 5V
 - A mono audio output can be obtained by carefully timing the pulses and relying on the speakers physical filtering properties
 - Sound output is typically very low quality
 - Generally used in soundtracks of many classic video games

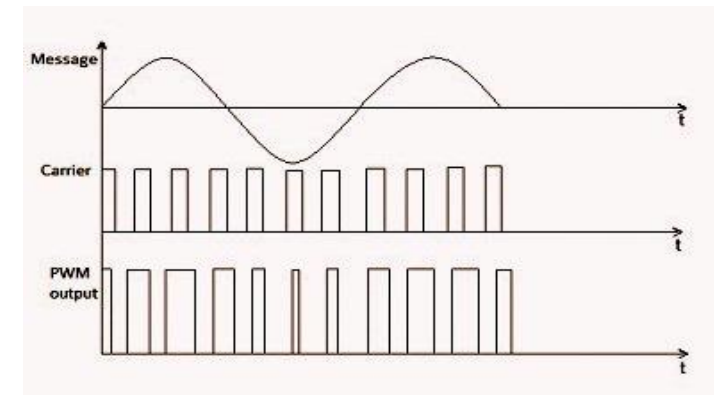


PCM to PWM

- ▶ PCM samples are taken at a regular clock interval
 - ▶ These signals represent a value between 0 and 255
- ▶ Using a clock we can convert these values into the appropriate duty cycles
- ▶ These duty cycles are converted to the duration at which the pulses are output in the PWM signal
- ▶ These varying pulse widths are what determine the voltage output which in turn creates the sound output from the speaker



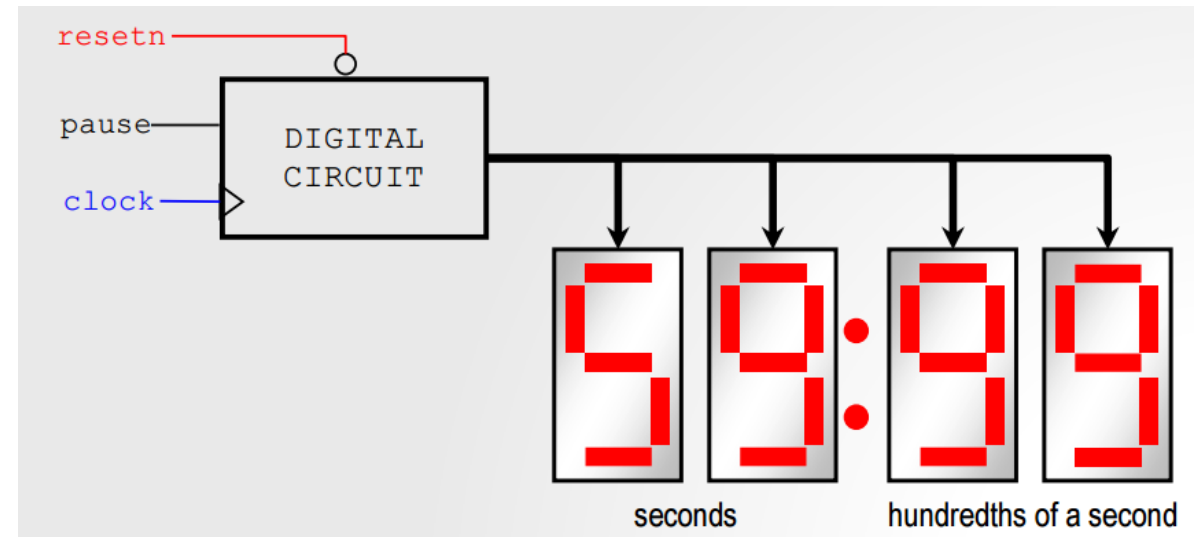
PCM



PWM

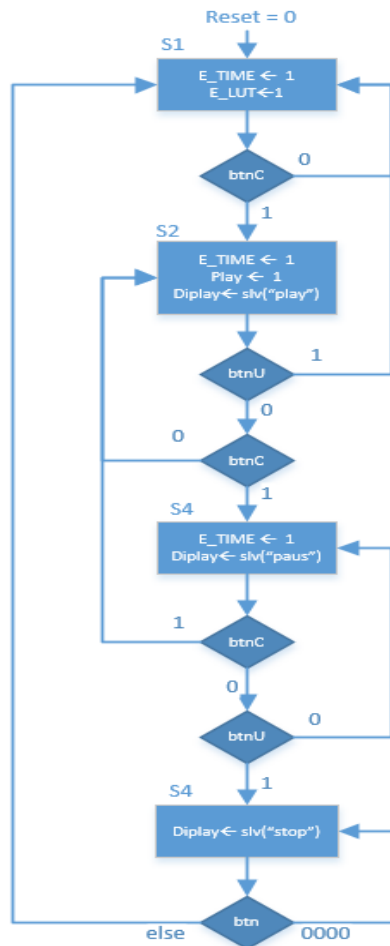
Timer (Seven Segment Display)

- ◆ Interfaced to keep track of the total time for which each sound plays
- ◆ Starts counting when the SD PCM data transfers
- ◆ Uses the PCM output of the Microprocessor unit to know when to stop based on the output
- ◆ First 4 displays used for name of track
- ◆ Last 4 displays used for timer

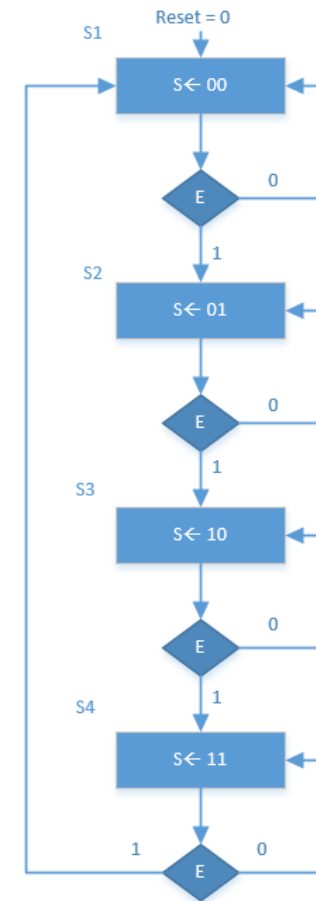


Display FSM's

7 Seg Displays



Timer



Further Design Applications

- ◆ Interfacing the SD card with the appropriate and understanding the overall file structure will lead to many different applications including:

- ◆ Loading multiple tracks
- ◆ Saving recorded tracks
- ◆ Looping multiple sounds
- ◆ Audio Amplification

