

Conductor Hero

Natalie Cheung, Yuta Kuboyama, Edgar Twigg

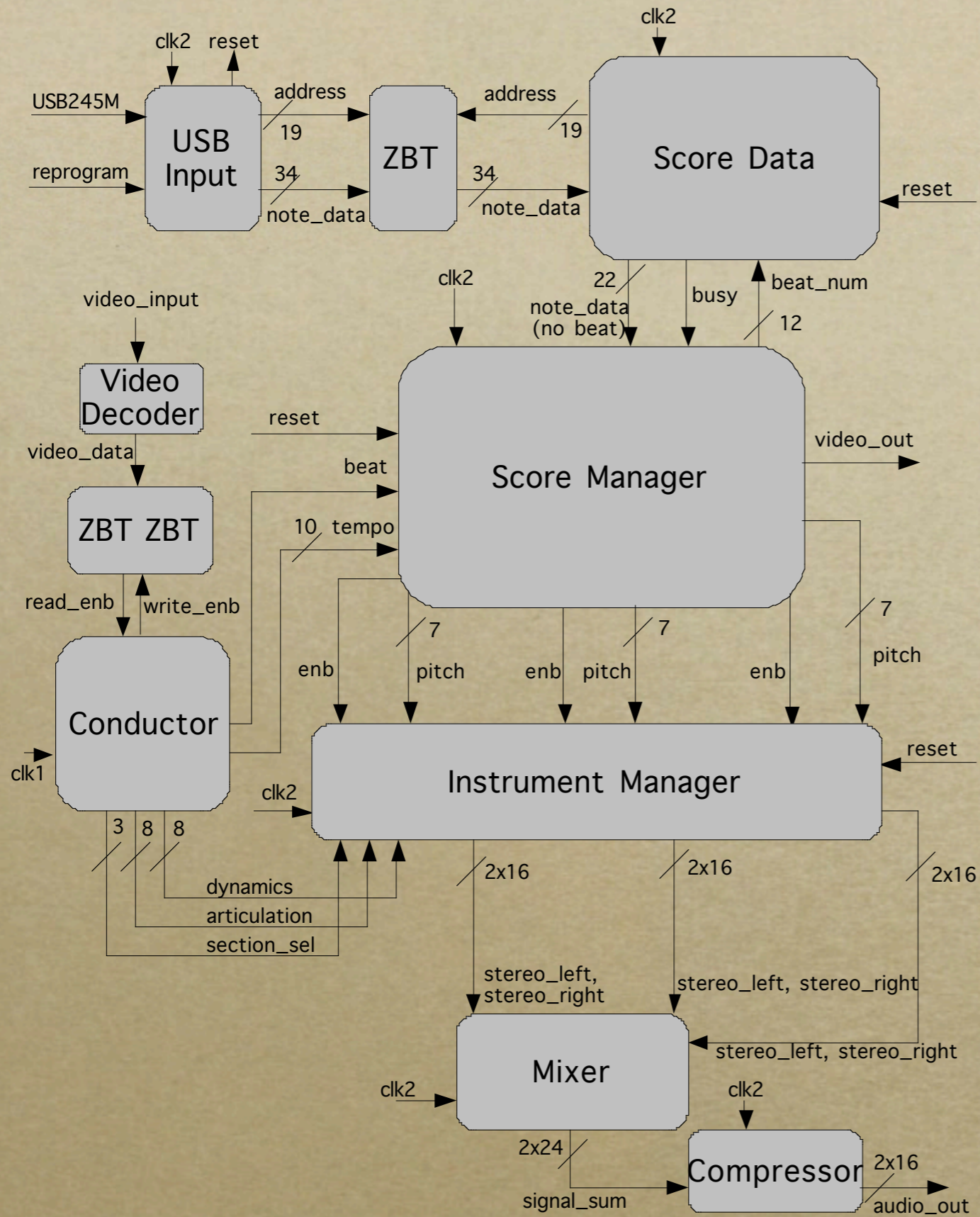
Introduction

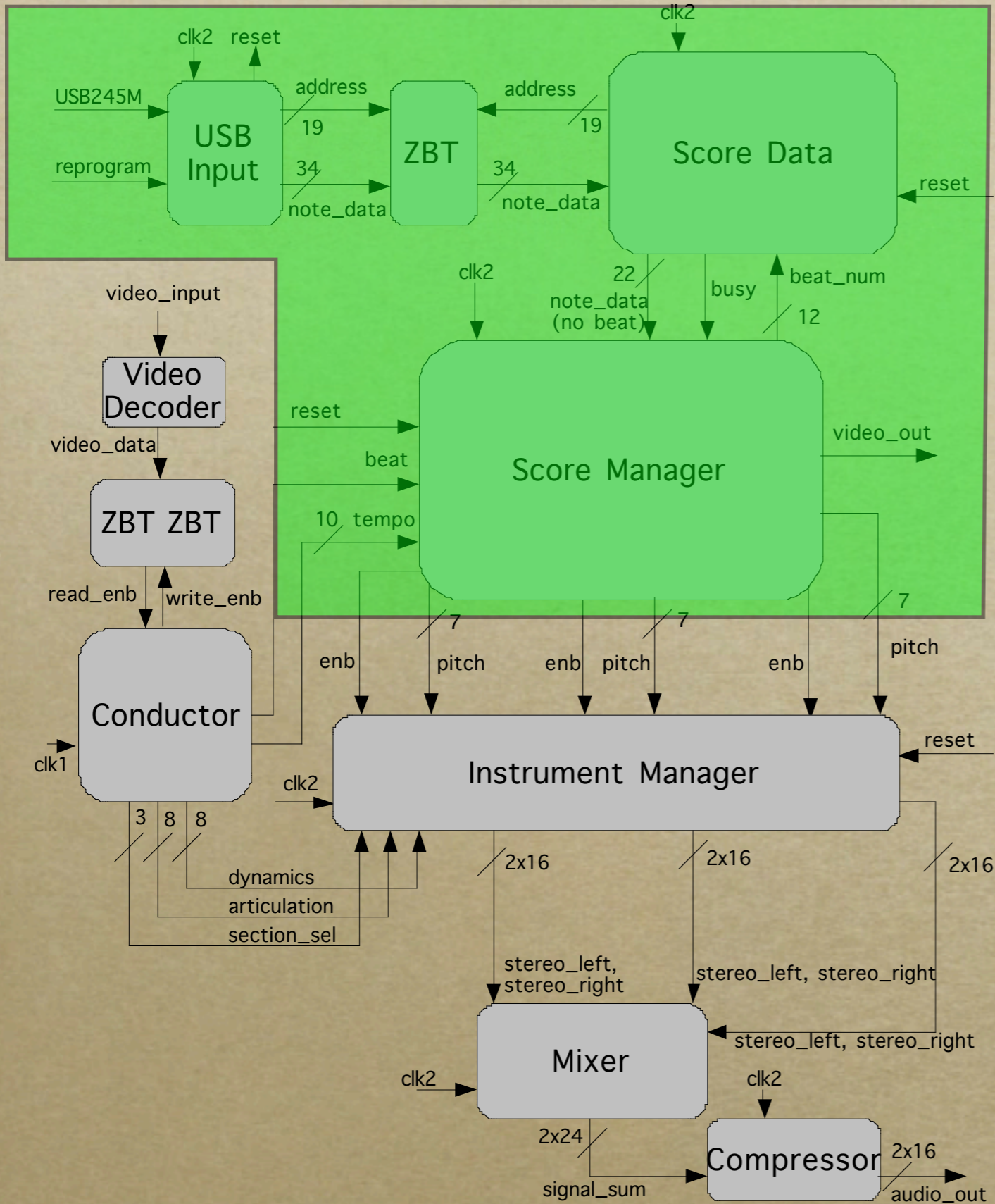


Overview

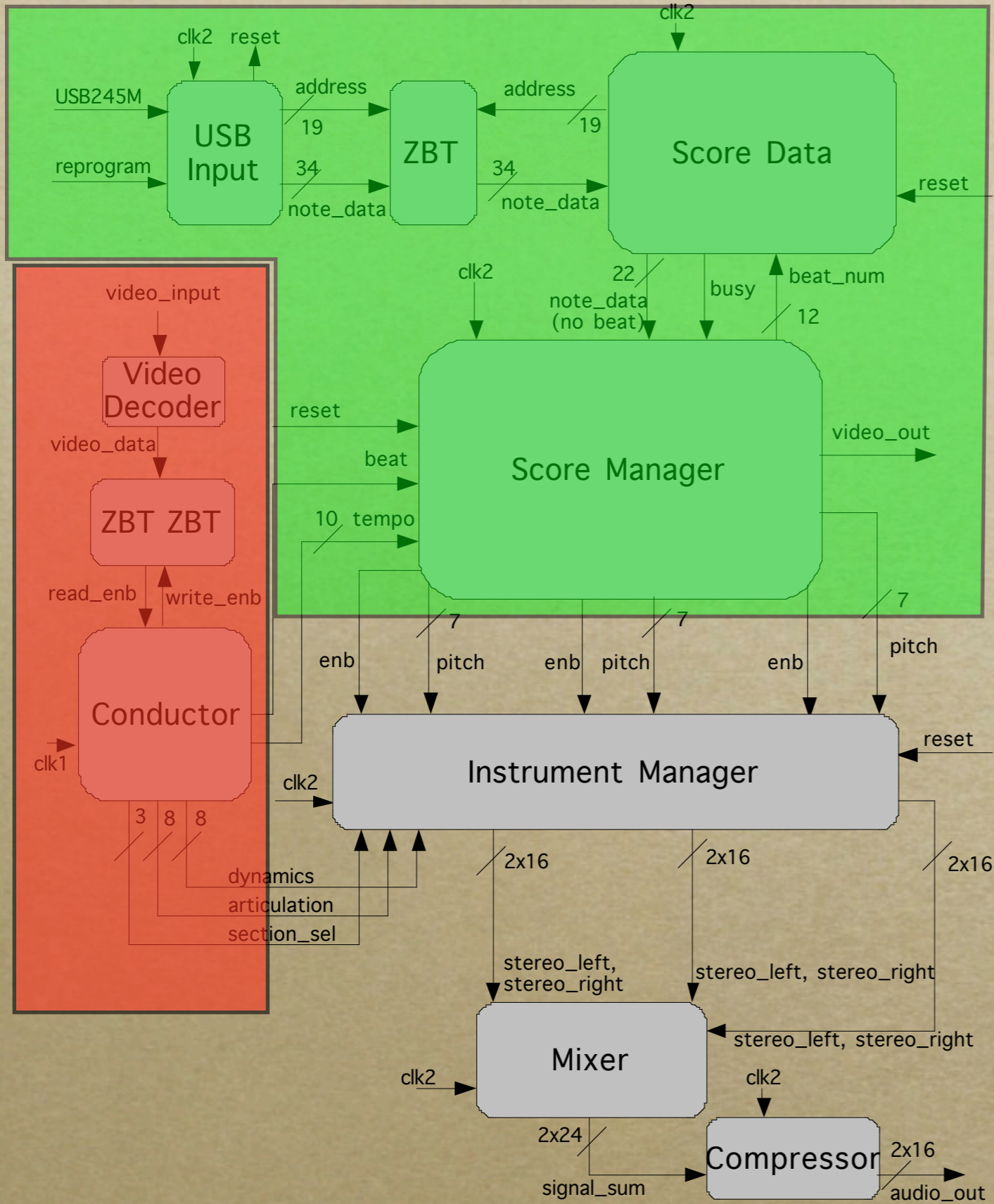
*Goal: Conduct an Orchestra based on
Hand Movements*

- *Score transferred through a USB Port*
- *User conducts the entire orchestra and smaller sections*
- *Output orchestra music and graphical representation of the orchestra on the screen*

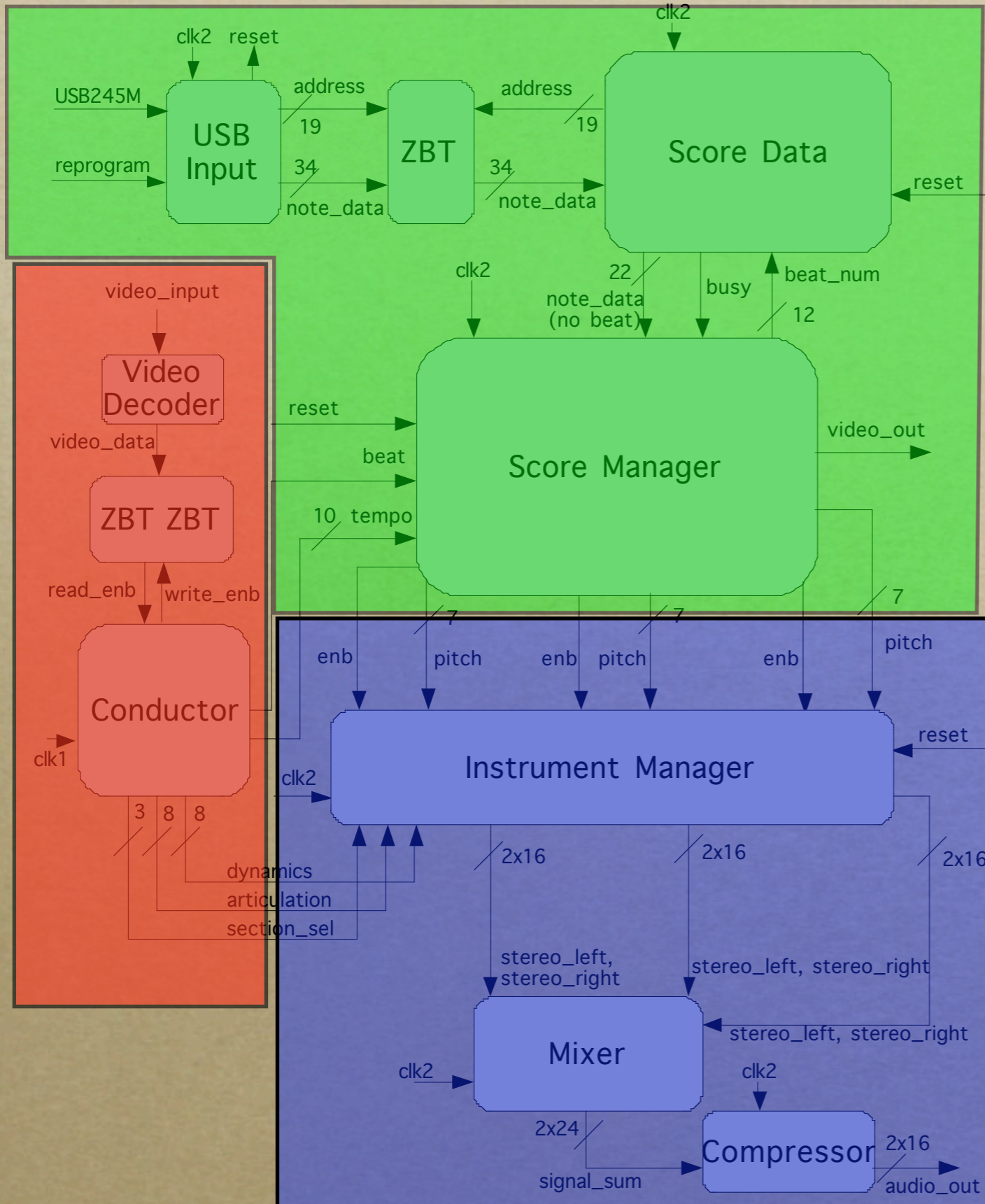




- *Score Management*
 - *Music data handling*
 - *Takes numeric user signals and score data to determine what instruments play when*

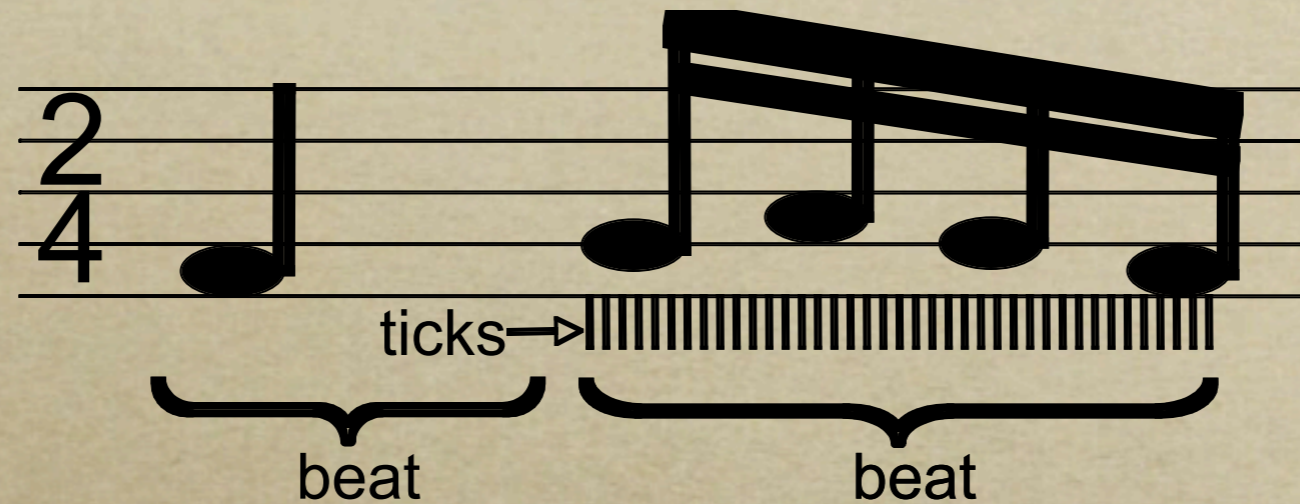


- *Score Management*
 - *Music data handling*
 - *Takes numeric user signals and score data to determine what instruments play when*
- *Conductor*
 - *User Interface*
 - *Generates numeric signals from user movement*



- *Score Management*
 - *Music data handling*
 - *Takes numeric user signals and score data to determine what instruments play when*
- *Conductor*
 - *User Interface*
 - *Generates numeric signals from user movement*
- *Sound Synthesis*
 - *Given on/off and pitch*
 - *Generates actual sound*

Time



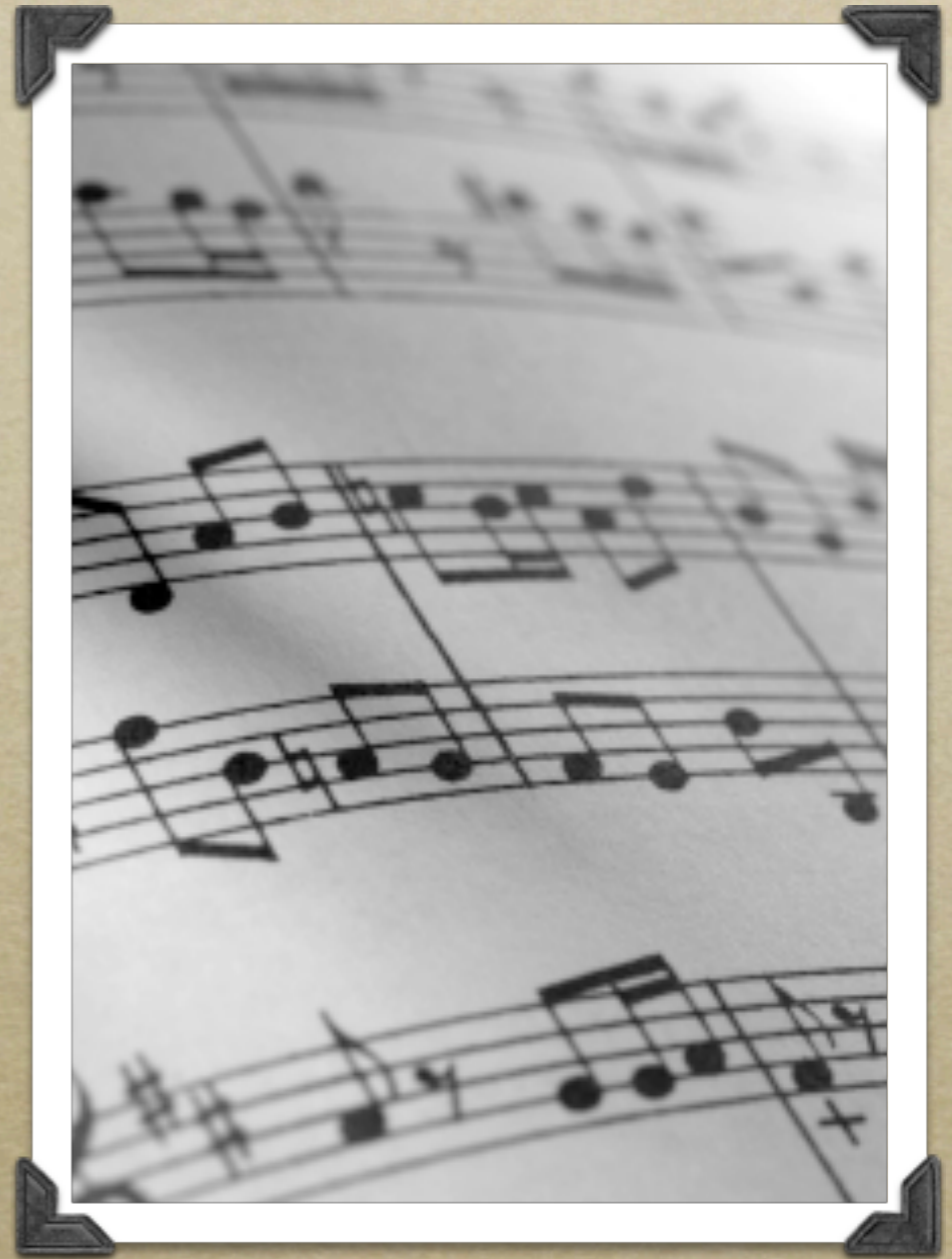
- *128 ticks per beat*
- *At 50 bpm a tick lasts for 9.4ms*
 - *humans can detect differences of ~20-30ms*
- *Uneven note durations (e.g. triplets) won't sound choppy*

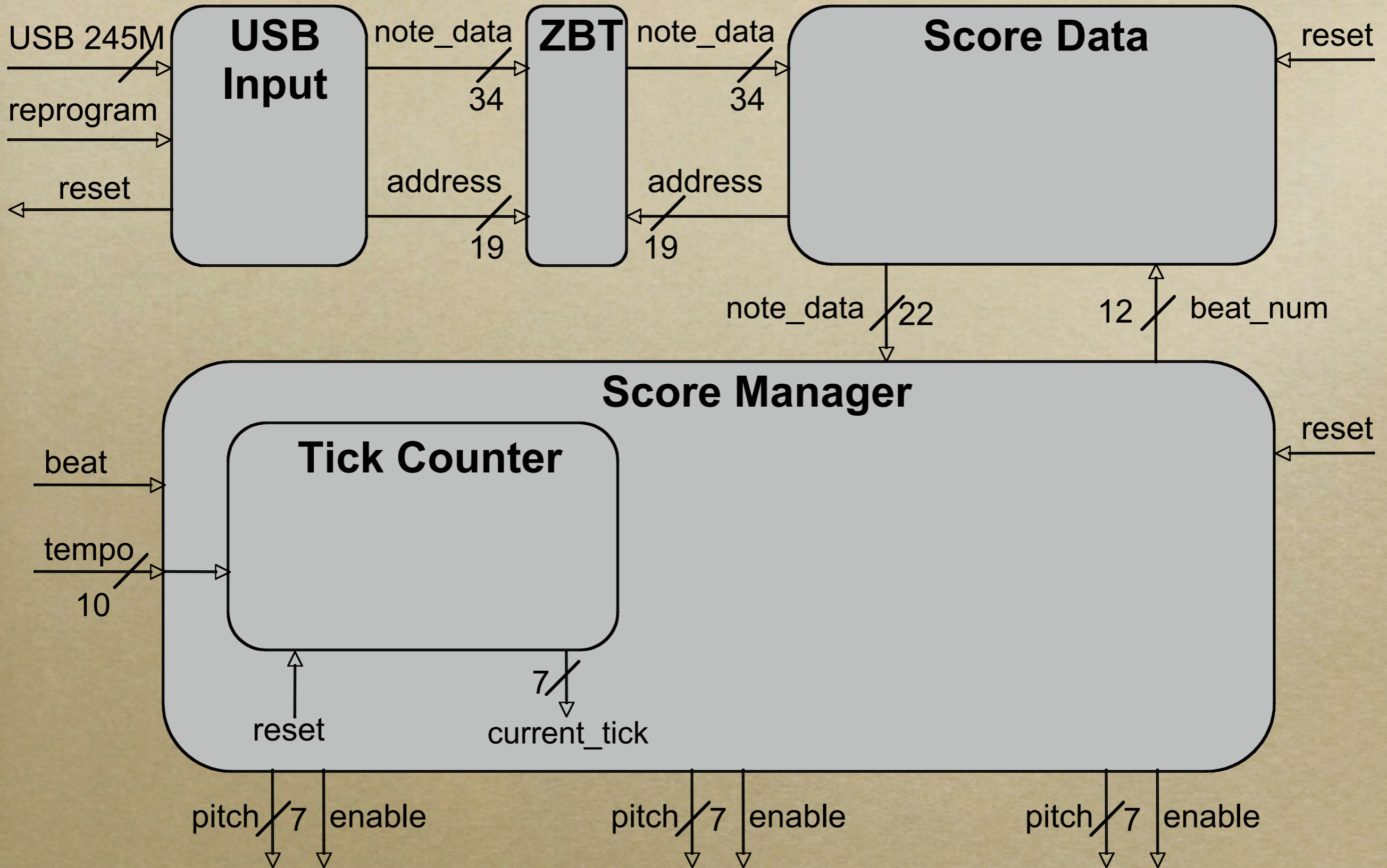
NoteData Format

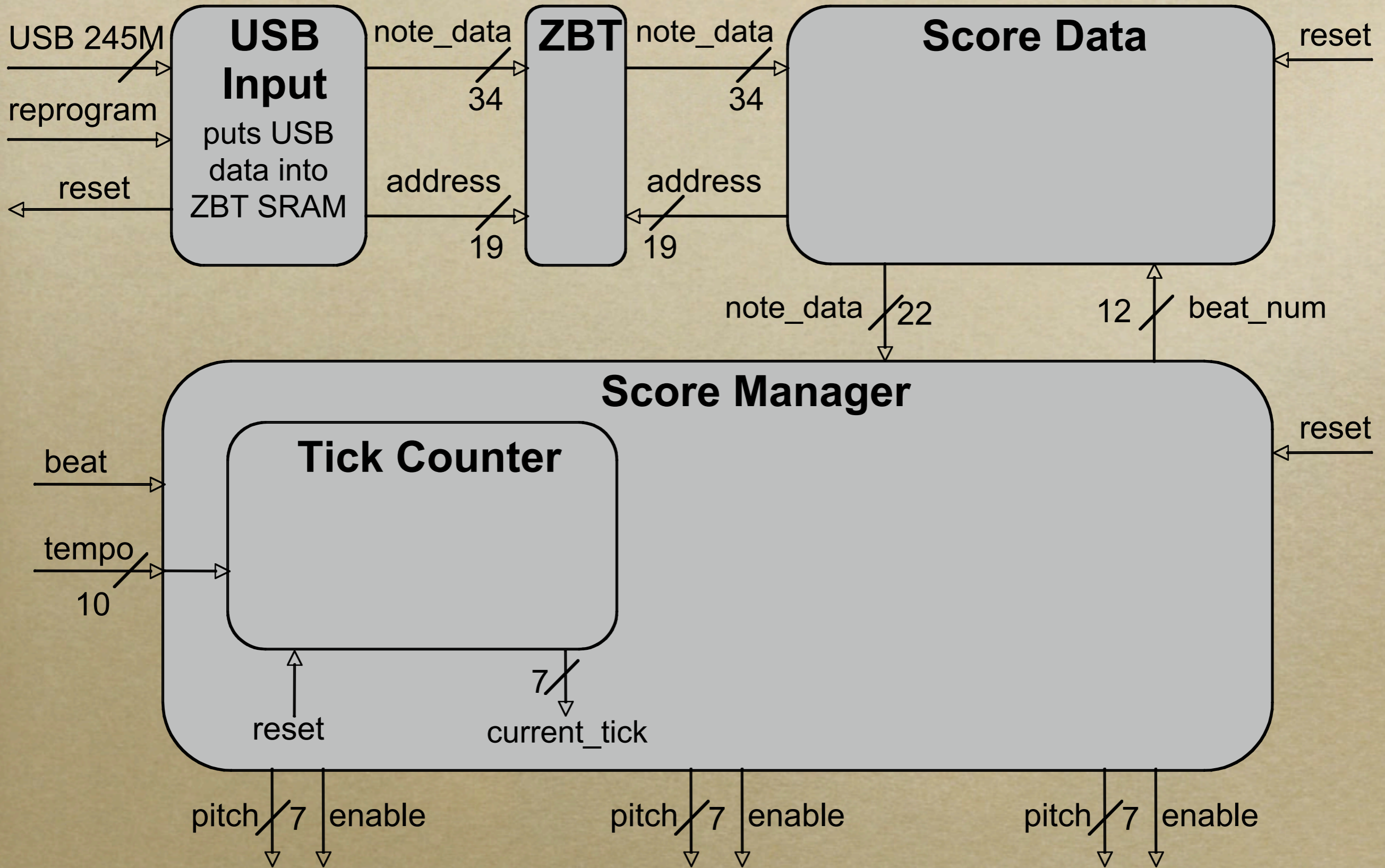
<i>Start Beat</i>	<i>Start Tick</i>	<i>Instrument</i>	<i>Pitch</i>	<i>Enable</i>
<i>0-11 (0-4095)</i>	<i>12-18 (0-127)</i>	<i>19-25 (0-127)</i>	<i>26-32 (0-127)</i>	<i>33 (0-1)</i>

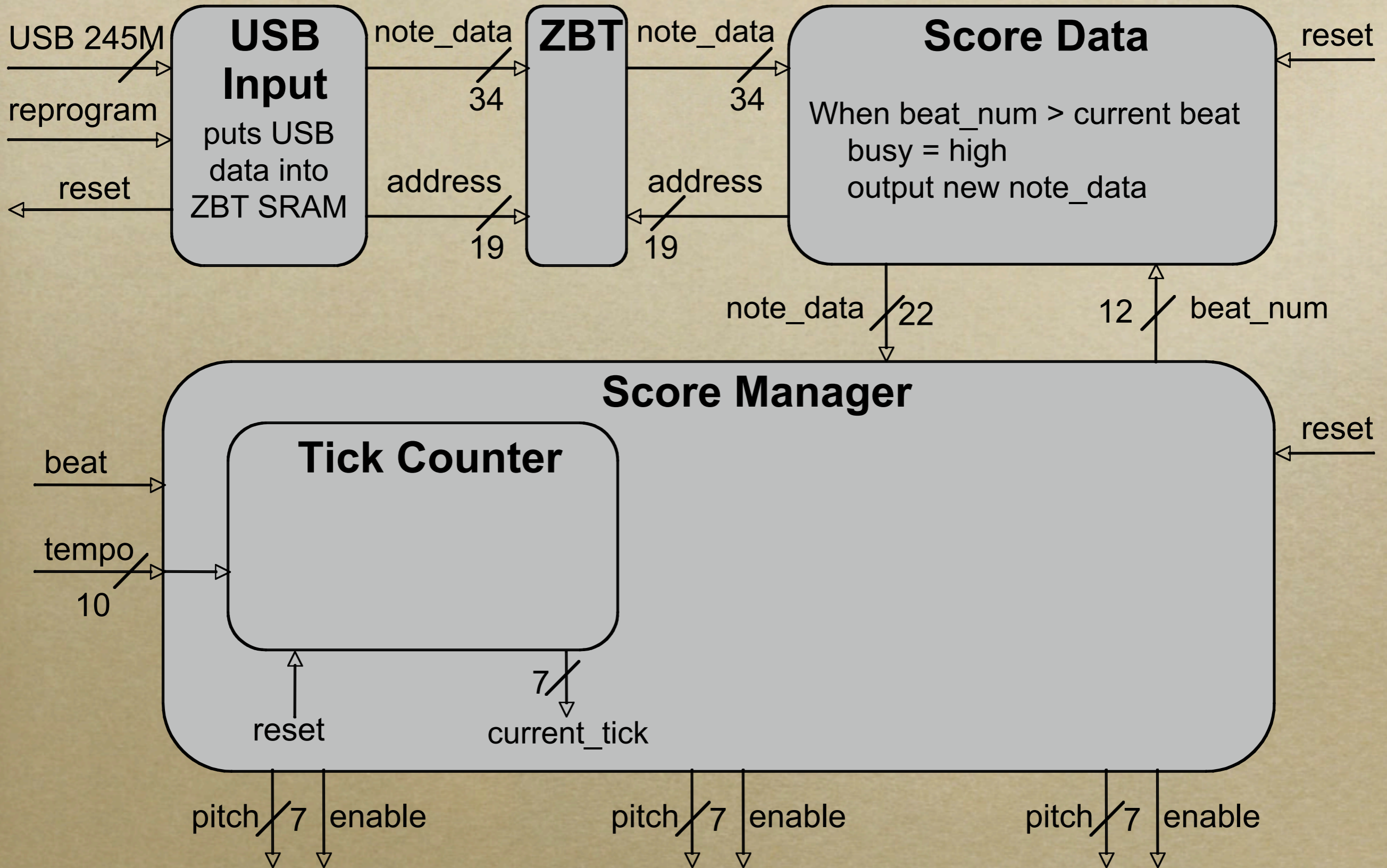
- *Each NoteData takes up one 36 bit wide SRAM location*
- *Two NoteData events per note (on and off) = 72 bits per note*
- *500 Measures in 4/4, average of 15 notes per beat = 2.06Mb*

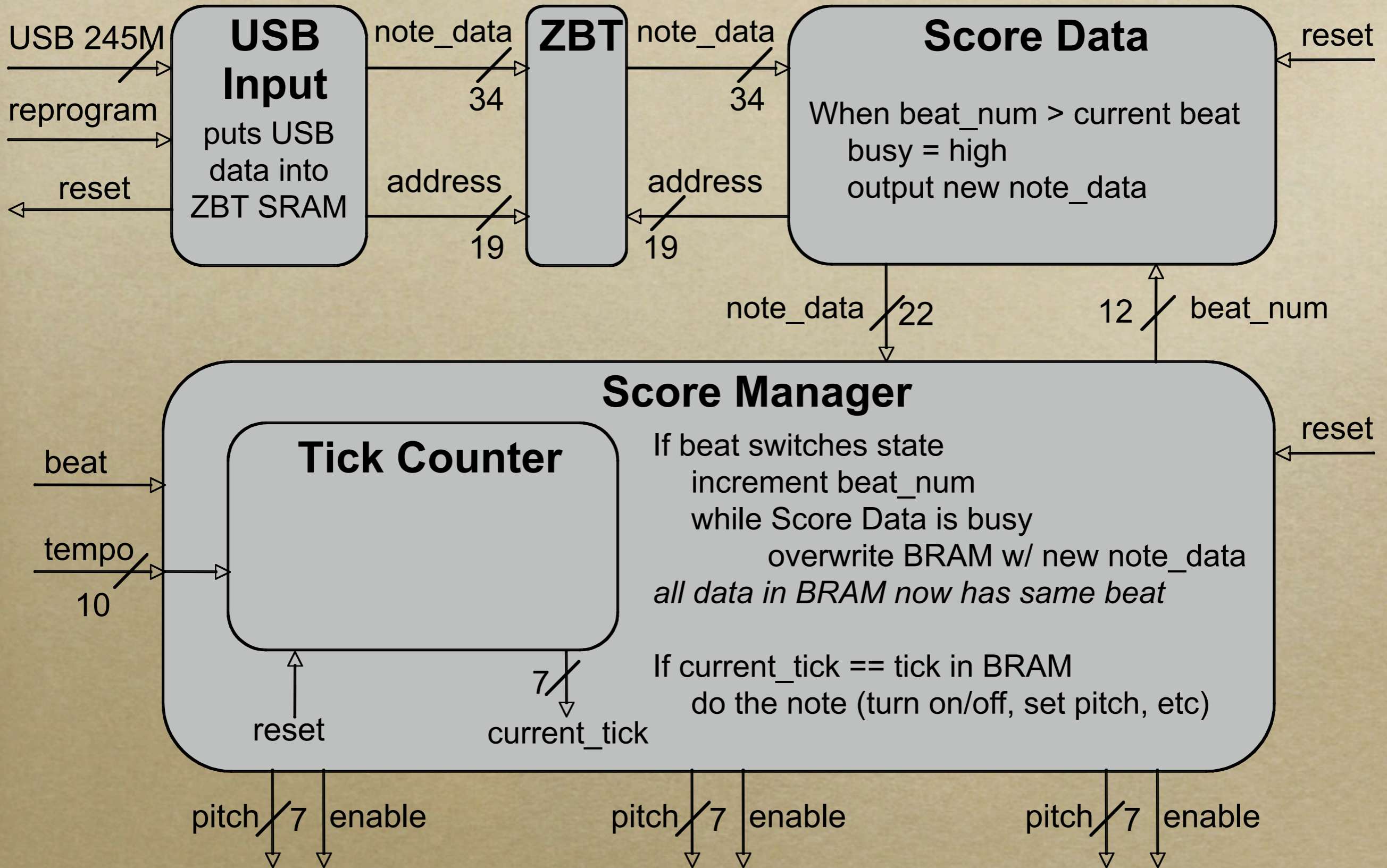
Score Management

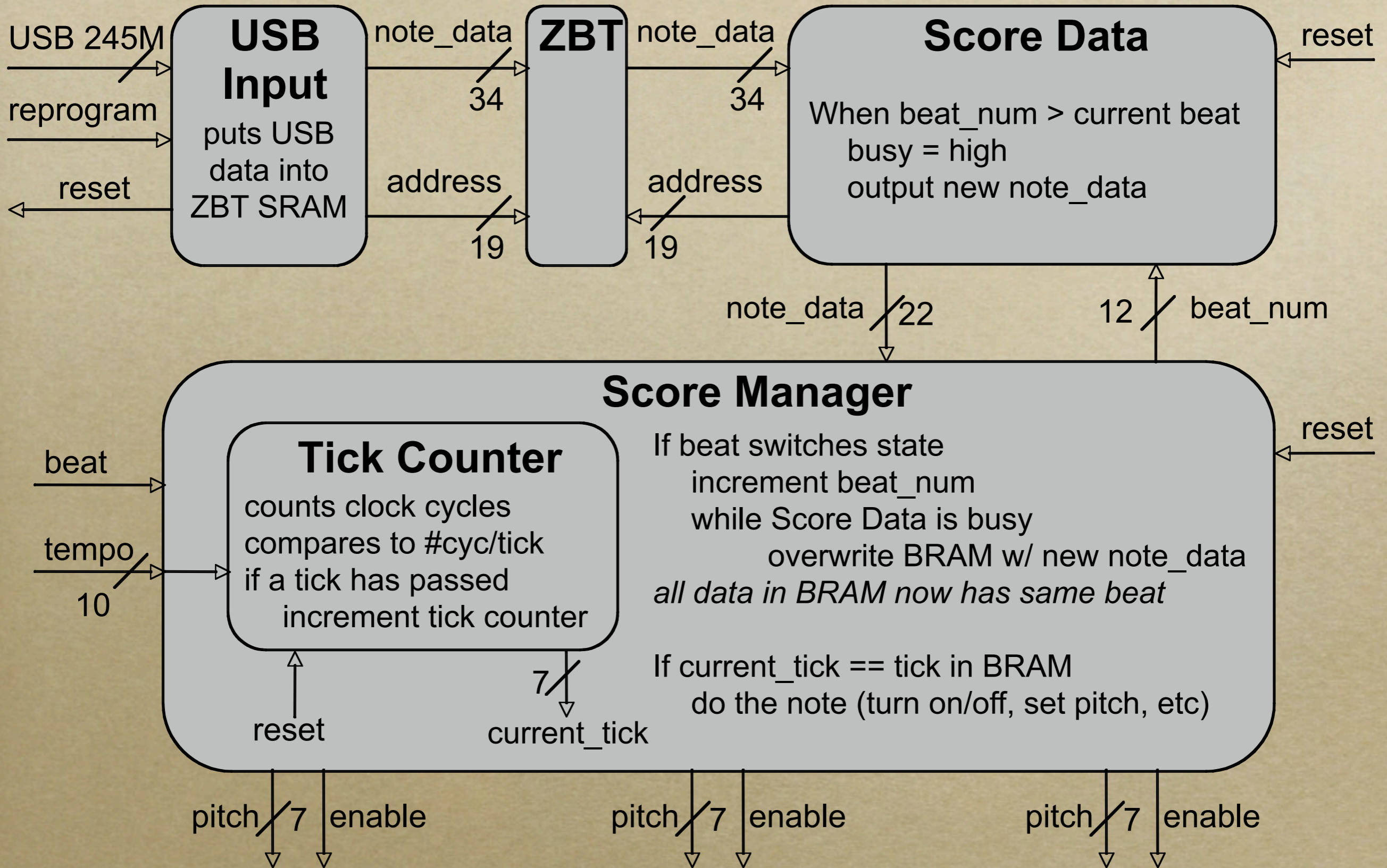












USB Input (Data Reading Timings)

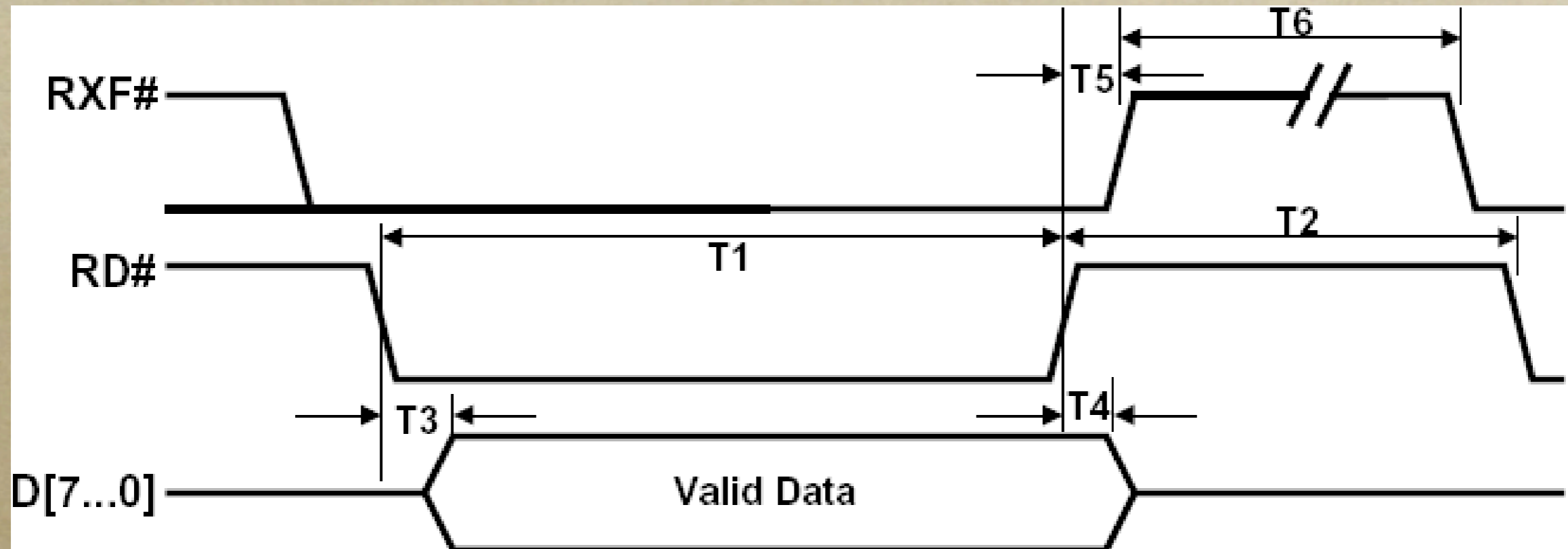
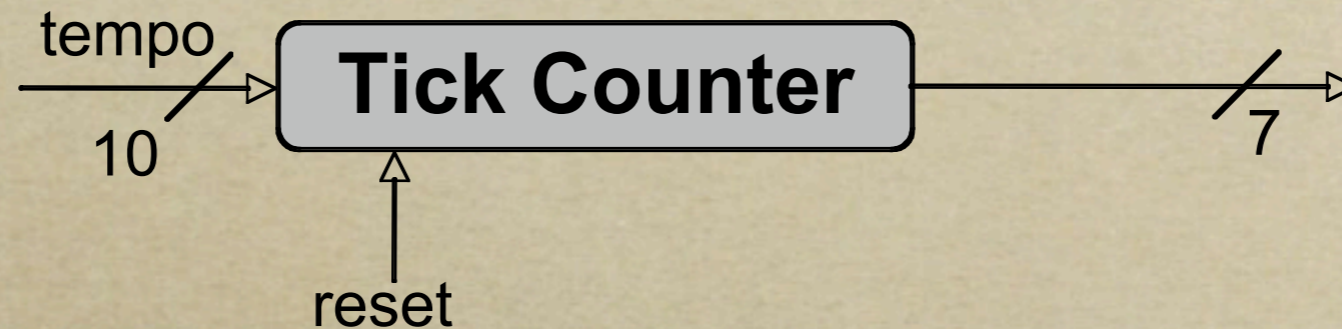


Table 4 - FIFO Read Cycle Timings

Time	Description	Min	Max	Unit
T1	RD Active Pulse Width	50		ns
T2	RD to RD Pre-Charge Time	50 + T6		ns
T3	RD Active to Valid Data*	20	50	ns
T4	Valid Data Hold Time from RD Inactive*	0		ns
T5	RD Inactive to RXF#	0	25	ns
T6	RXF Inactive After RD Cycle	80		ns

- excerpt from FTDI
UM245R USB-Parallel
FIFO Development
Module Datasheet

Tick Counter



- *Counts the # of clock cycles that have elapsed since the last tick*
- *Calculates the # of clock cycles that should elapse before the next tick based on the tempo*
- *Increments the current tick when count = calculated*

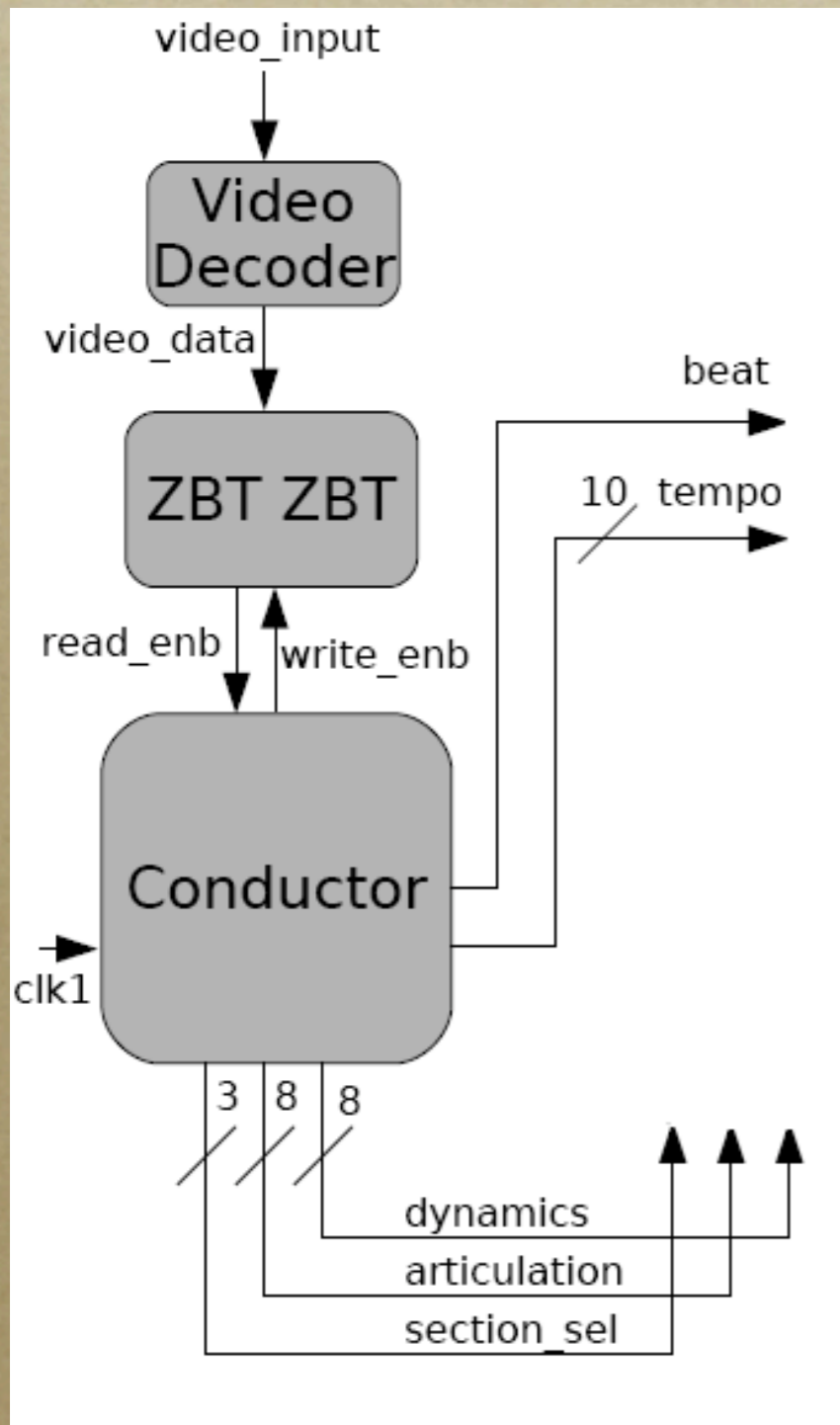
$$\frac{\text{tempo}}{4} \frac{\text{beats}}{\text{min}} \cdot 128 \frac{\text{tick}}{\text{min}} \cdot \frac{1}{60} \frac{\text{min}}{\text{sec}} = \text{tempo} \cdot \frac{8}{15} \frac{\text{tick}}{\text{sec}}$$

$$\frac{1}{\text{tempo}} \cdot \frac{15}{8} \frac{\text{sec}}{\text{tick}} \cdot 27,000,000 \frac{\text{clk}}{\text{sec}} = \frac{50,625,000}{\text{tempo}} \frac{\text{clk}}{\text{tick}}$$

Conductor



Conductor Module

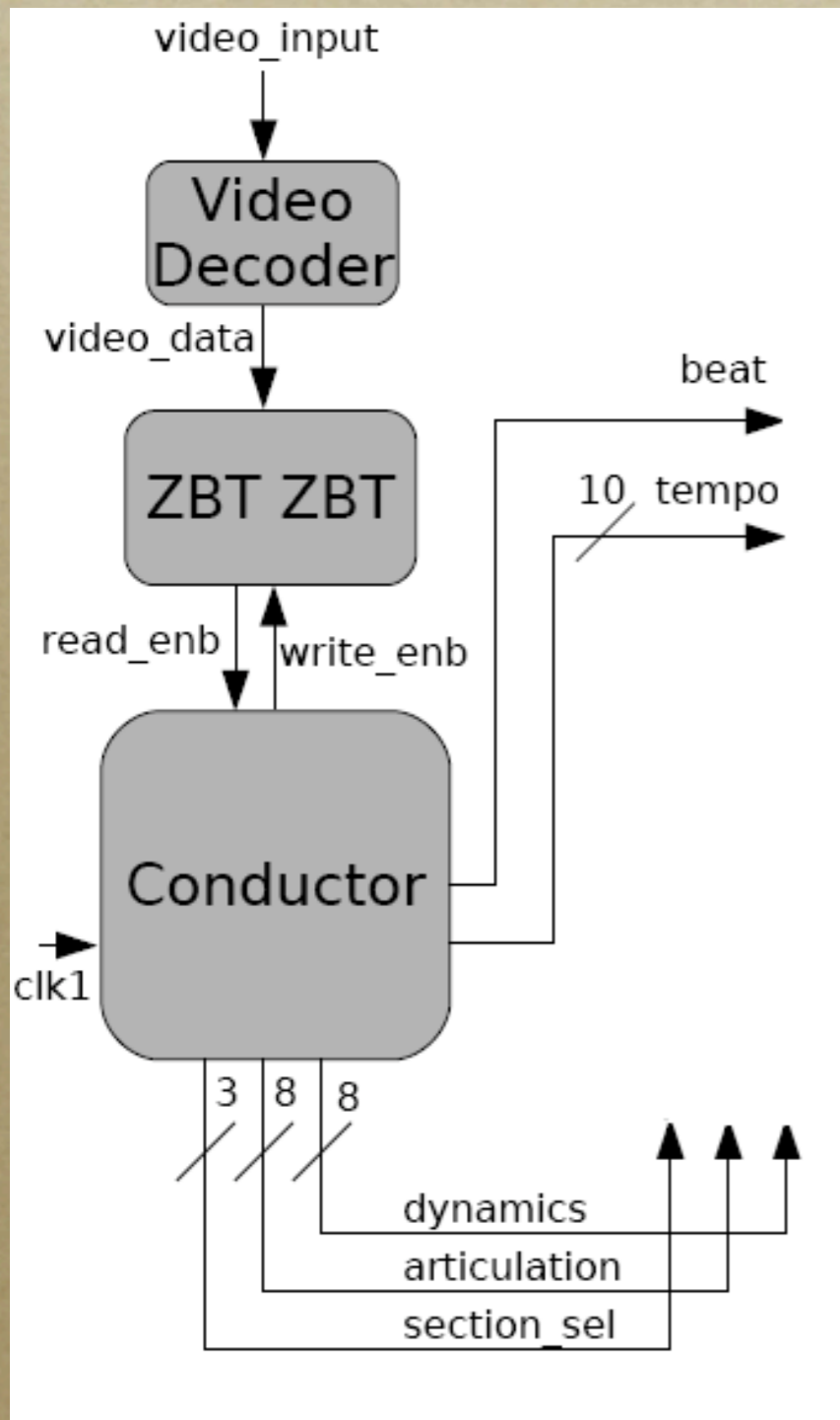


Video Decoder

- *Detect LED using thresholds*
- *Reject other colors*
- *Store data into ZBT*



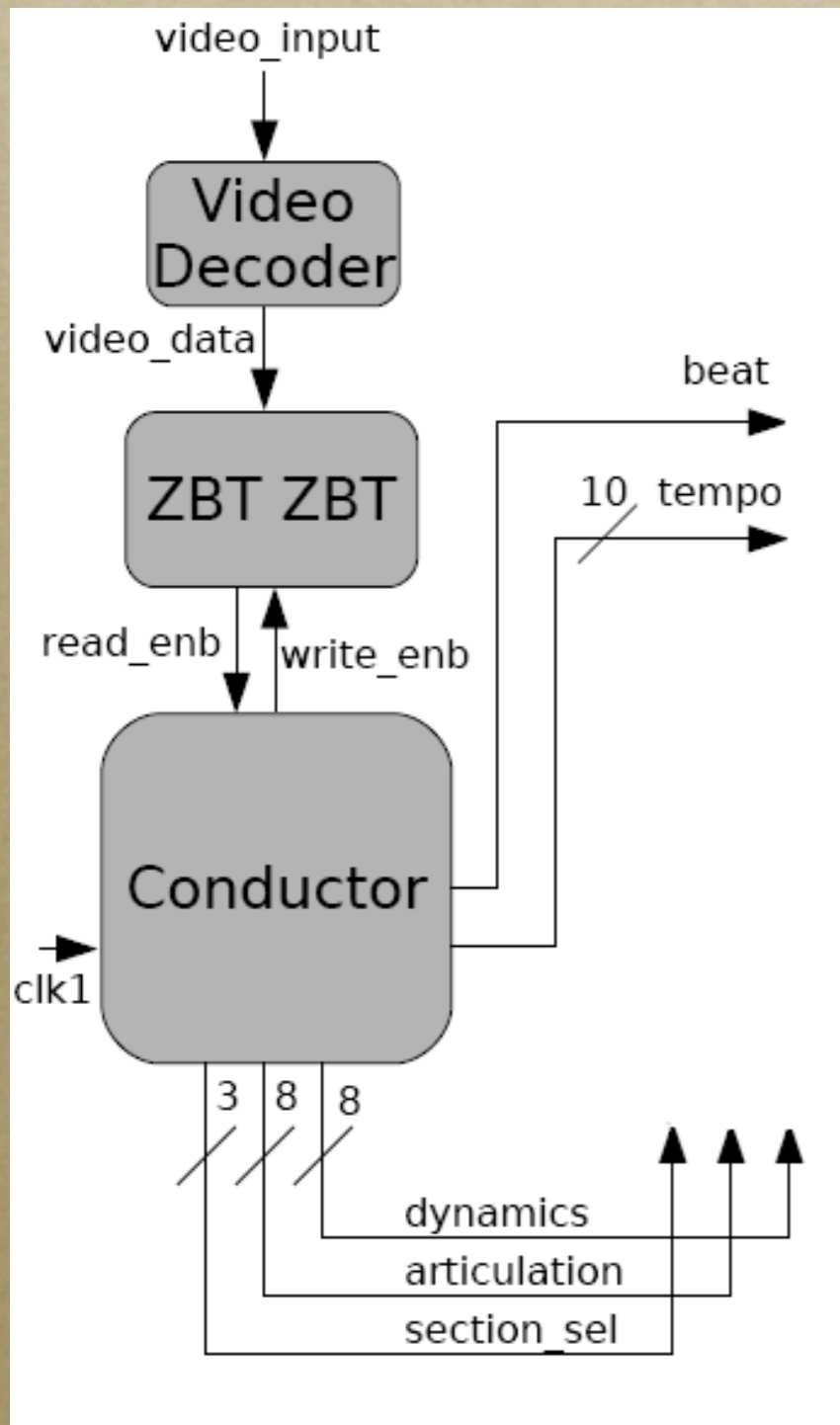
Conductor Module



ZBT

- *Each frame*
 - *ZBT1 collects information from video_data*
 - *Conductor module reads from ZBT2*
- *Every frame*
 - *ZBT1 <= ZBT2*
 - *ZBT2 <= ZBT1*

Conductor Module



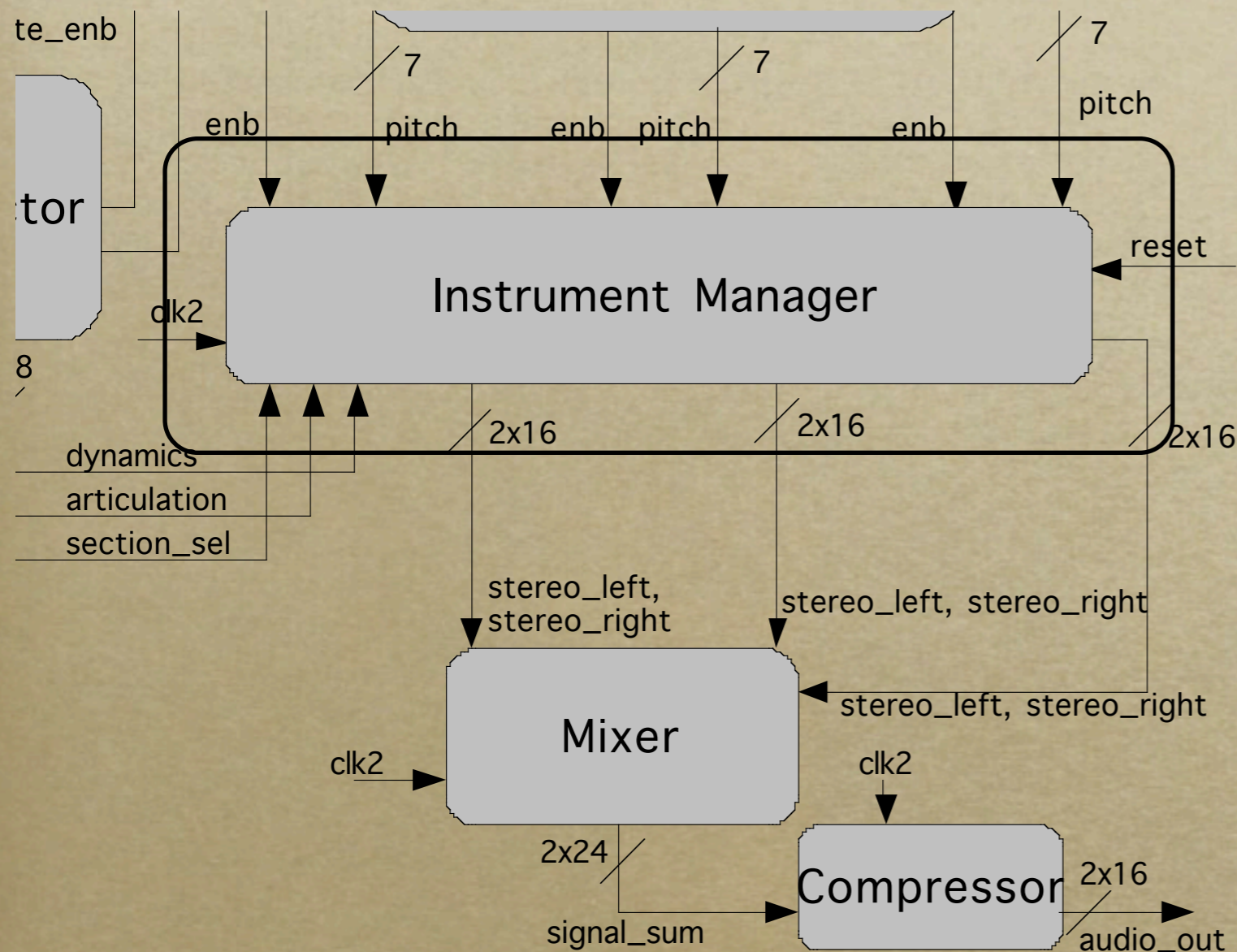
Conductor

- *Determine if there are 2 LEDs (conduct the whole orchestra)*
- *Determine position of LED*
- *If position of LED remains the same for a certain number of frames (based on a counter), the next time the LED moves to a different position, the beat enable is high*
- *Tempo is based on the velocity of the LED*

Sound Synthesis

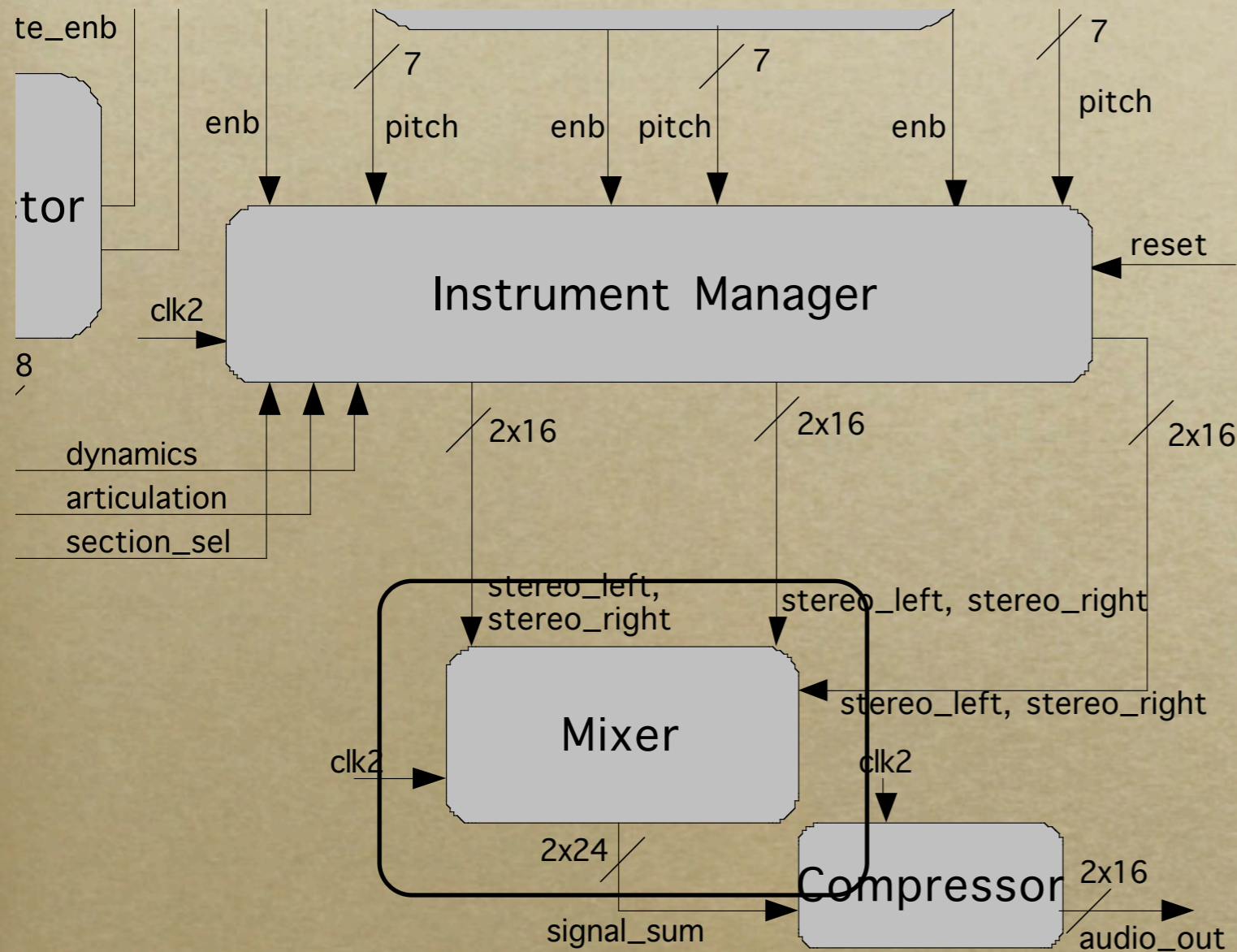


Instrument Manager



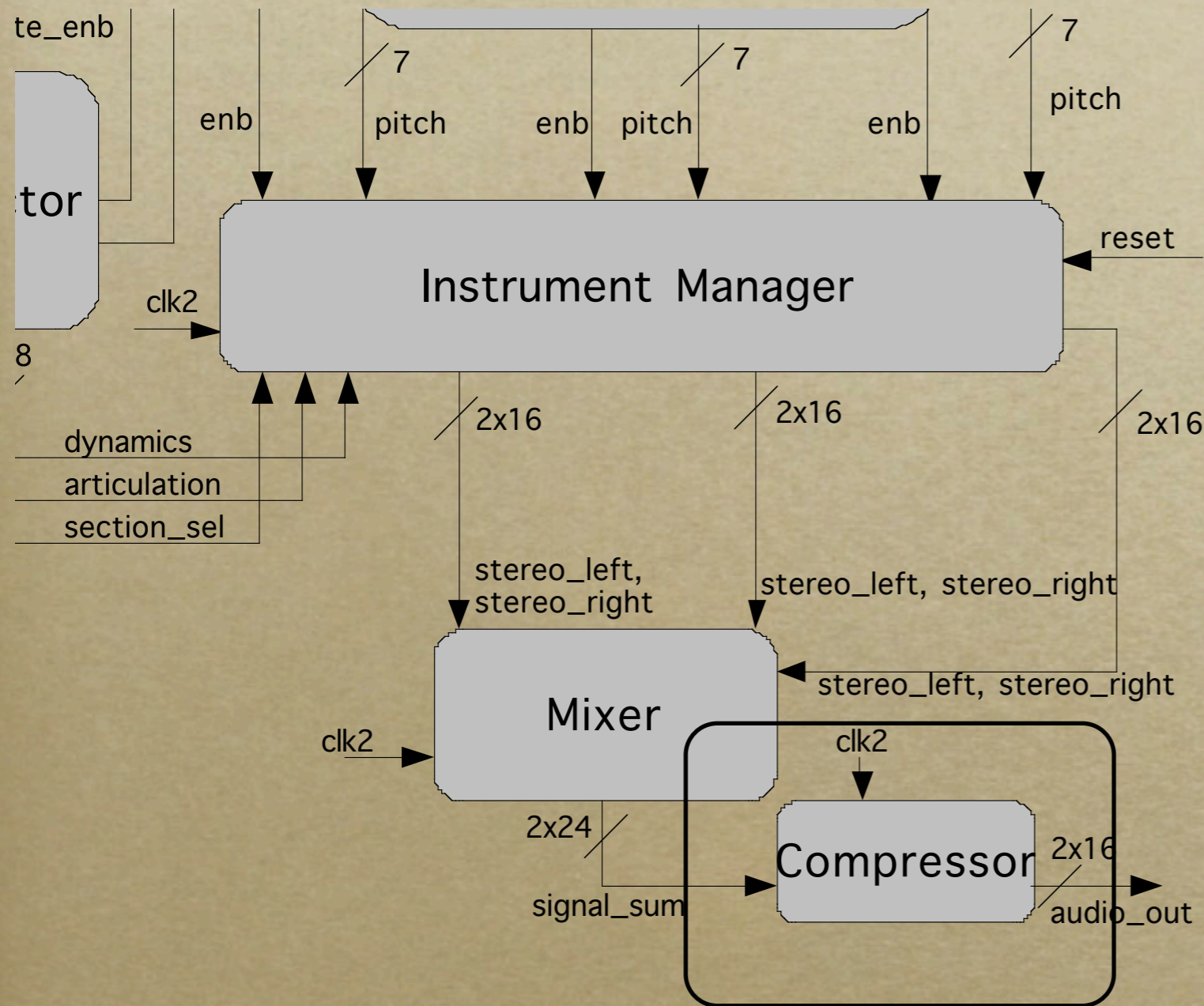
- Represents a section of instruments (strings, brass, etc), and produces audio signals for the instruments at specified pitches
- Consists of instrument parameters, signal processor, sample look-up table
- Updates the dynamics and articulation when the section selector matches
- Outputs a stereo signal representing the location in the orchestra

Mixer



- *Simply adds the left channel and right channel signals separately*
- *Produces a 24-bit stereo signal from adding 16-bit stereo signals*

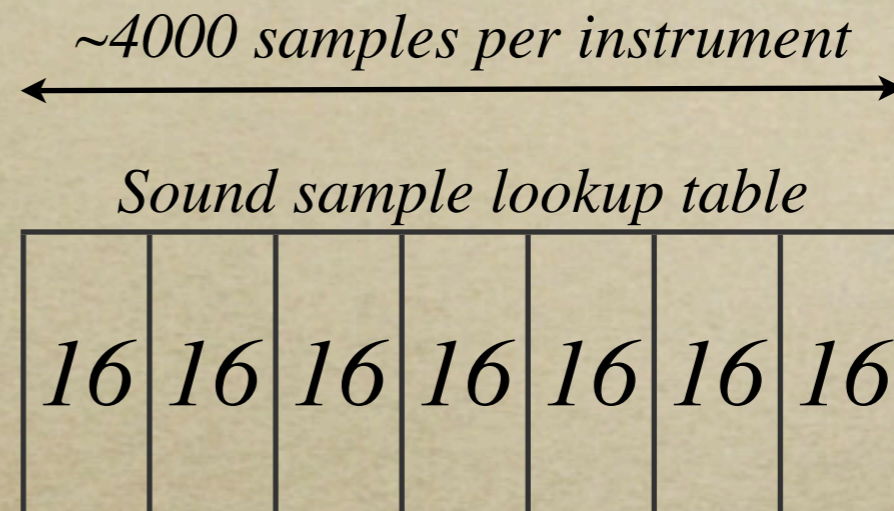
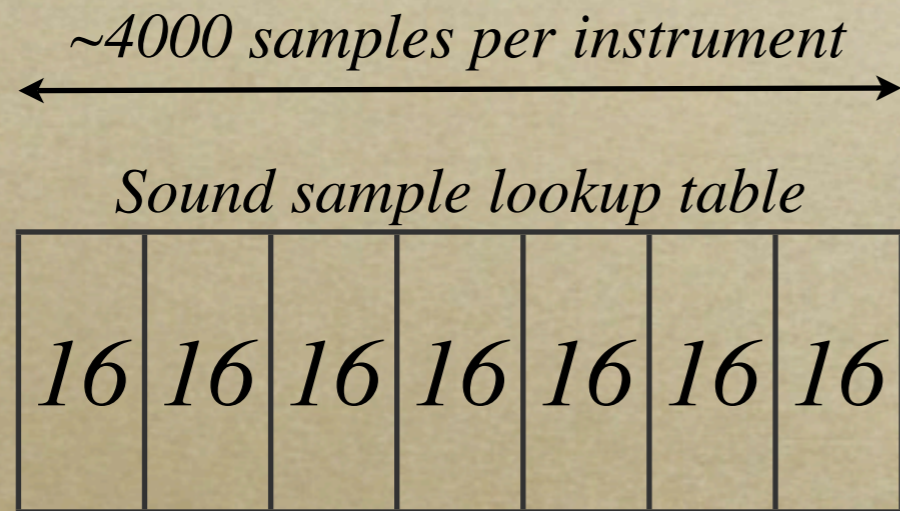
Compressor



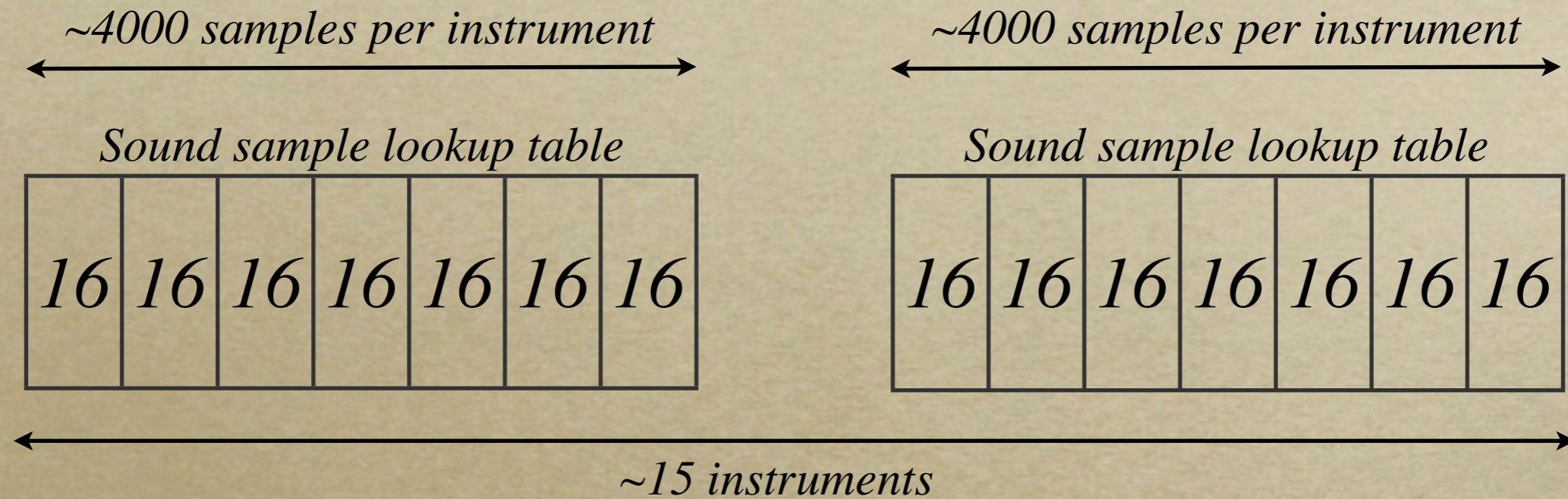
- *Balances the output and prevents clipping or getting too quiet*
- *Applies gain < 1 for signals that would clip*
- *Applies gain > 1 for signals that are too quiet*
- *Sends a 16-bit stereo signal to the DAC, then to the speakers*

Memory and Resources

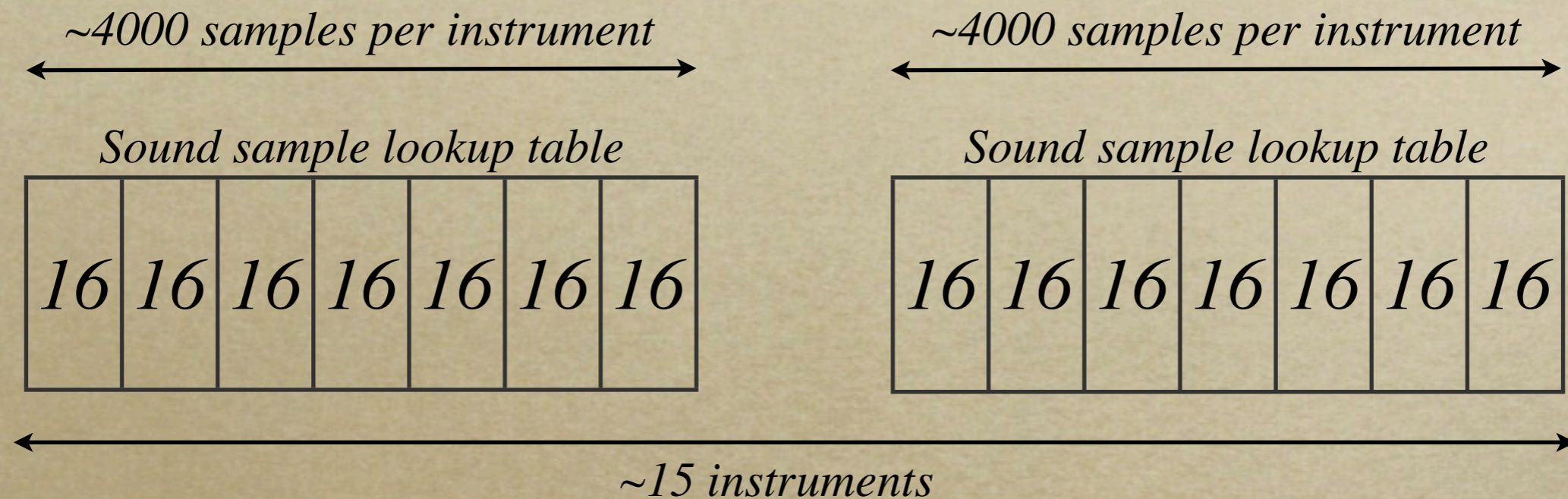
Memory and Resources



Memory and Resources

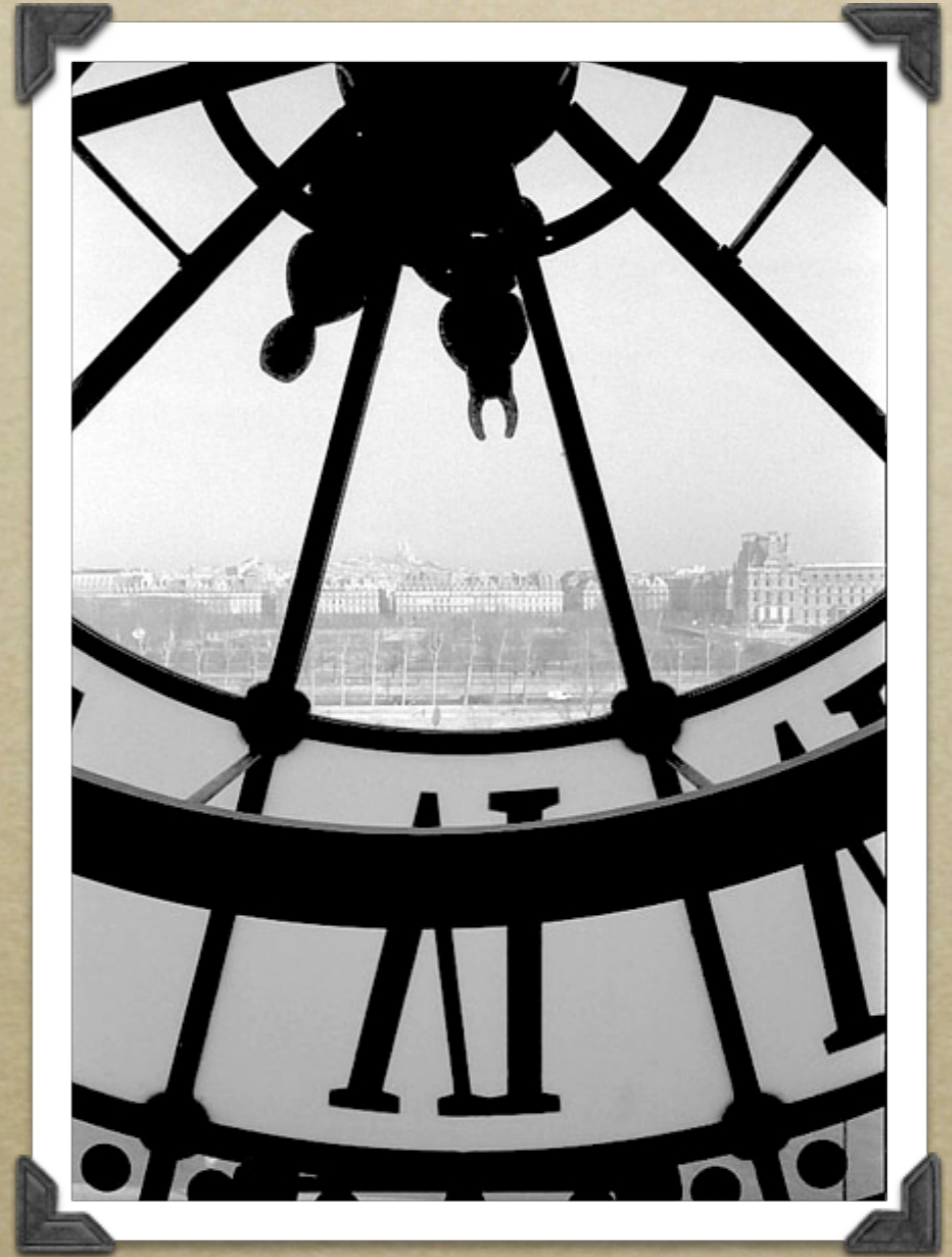


Memory and Resources



0.96Mbits out of 2.5Mbits on one FPGA

Finale



Timeline

	<i>Sun</i>	<i>Mon</i>	<i>Tues</i>	<i>Wed</i>	<i>Thurs</i>	<i>Fri</i>	<i>Sat</i>
<i>Nov</i>	11	12	13	14	15 <i>Today</i>	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	1
<i>Dec</i>	2	3	4	5	6	7	8
	9	10	11	12	13	14	15

Timeline

	<i>Sun</i>	<i>Mon</i>	<i>Tues</i>	<i>Wed</i>	<i>Thurs</i>	<i>Fri</i>	<i>Sat</i>
<i>Nov</i>	11	12	13	14	15 <i>Today</i>	16 <i>Detect LED Color</i> <i>Write USB Input Module</i> <i>Sample Audio & Playback</i>	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	1
<i>Dec</i>	2	3	4	5	6	7	8
	9	10	11	12	13	14	15

Timeline

	<i>Sun</i>	<i>Mon</i>	<i>Tues</i>	<i>Wed</i>	<i>Thurs</i>	<i>Fri</i>	<i>Sat</i>
<i>Nov</i>	11	12	13	14	15	16	17
						<i>Today</i>	<div style="background-color: #f8d7da; padding: 2px;"><i>Detect LED Color</i></div> <div style="background-color: #d4edda; padding: 2px;"><i>Write USB Input Module</i></div> <div style="background-color: #d1ecf1; padding: 2px;"><i>Sample Audio & Playback</i></div>
	18	19	20	21	22	23	24
<div style="background-color: #f8d7da; padding: 2px;"><i>Collect data and determine Beat and Tempo</i></div> <div style="background-color: #d4edda; padding: 2px;"><i>Write Score Data Module</i></div> <div style="background-color: #d1ecf1; padding: 2px;"><i>Playback audio at given pitch</i></div>							
	25	26	27	28	29	30	1
<i>Dec</i>	2	3	4	5	6	7	8
	9	10	11	12	13	14	15

Timeline

	Sun	Mon	Tues	Wed	Thurs	Fri	Sat
Nov	11	12	13	14	15 <i>Today</i>	16 <i>Detect LED Color</i>	17 <i>Write USB Input Module</i> <i>Sample Audio & Playback</i>
	18 <i>Collect data and determine Beat and Tempo</i>	19 <i>Write Score Data Module</i>	20 <i>Playback audio at given pitch</i>	21	22	23	24 <i>Thanksgiving Break</i>
	25	26	27	28	29	30	1
Dec	2	3	4	5	6	7	8
	9	10	11	12	13	14	15

Timeline

	Sun	Mon	Tues	Wed	Thurs	Fri	Sat
Nov	11	12	13	14	15 <i>Today</i>	16 <i>Detect LED Color</i>	17 <i>Write USB Input Module</i> <i>Sample Audio & Playback</i>
	18 <i>Collect data and determine Beat and Tempo</i>	19 <i>Write Score Data Module</i>	20 <i>Playback audio at given pitch</i>	21	22	23	24 <i>Thanksgiving Break</i>
	25	26 <i>Create Conductor Module for whole orchestra</i>	27 <i>Write Score Manager</i>	28 <i>Playback audio with Dynamics and Articulation Parameters</i>	29	30	1
Dec	2	3	4	5	6	7	8
	9	10	11	12	13	14	15

Timeline

	Sun	Mon	Tues	Wed	Thurs	Fri	Sat	
Nov	11	12	13	14	15	16	17	
					<i>Today</i>	<i>Detect LED Color</i>	<i>Write USB Input Module</i>	<i>Sample Audio & Playback</i>
	18	19	20	21	22	23	24	
	<i>Collect data and determine Beat and Tempo</i>							
	<i>Write Score Data Module</i>							
	<i>Playback audio at given pitch</i>					<i>Thanksgiving Break</i>		
	25	26	27	28	29	30	1	
	<i>Create Conductor Module for whole orchestra</i>							
	<i>Write Score Manager</i>							
	<i>Playback audio with Dynamics and Articulation Parameters</i>					<i>Integration</i>		
	Dec	2	3	4	5	6	7	8
<i>Integration</i>								
9		10	11	12	13	14	15	

Timeline

	Sun	Mon	Tues	Wed	Thurs	Fri	Sat	
Nov	11	12	13	14	15 <i>Today</i>	16 <i>Detect LED Color</i> <i>Write USB Input Module</i> <i>Sample Audio & Playback</i>	17	
	18 <i>Collect data and determine Beat and Tempo</i> <i>Write Score Data Module</i> <i>Playback audio at given pitch</i>	19	20	21	22	23	24	
						<i>Thanksgiving Break</i>		
	25	26 <i>Create Conductor Module for whole orchestra</i> <i>Write Score Manager</i> <i>Playback audio with Dynamics and Articulation Parameters</i>	27	28	29	30	1	
						<i>Integration</i>		
Dec	2	3	4	5	6	7	8	
					<i>Finishing Touches</i>			
	9	10	11	12	13	14	15	

Timeline

	Sun	Mon	Tues	Wed	Thurs	Fri	Sat
Nov	11	12	13	14	15	16	17
					<i>Today</i>	<i>Detect LED Color</i>	
						<i>Write USB Input Module</i>	
						<i>Sample Audio & Playback</i>	
	18	19	20	21	22	23	24
	<i>Collect data and determine Beat and Tempo</i>						
	<i>Write Score Data Module</i>						
	<i>Playback audio at given pitch</i>						
						<i>Thanksgiving Break</i>	
	25	26	27	28	29	30	1
	<i>Create Conductor Module for whole orchestra</i>						
	<i>Write Score Manager</i>						
<i>Playback audio with Dynamics and Articulation Parameters</i>							
					<i>Integration</i>		
Dec	2	3	4	5	6	7	8
						<i>Add orchstra sections and graphics</i>	
						<i>Write MIDI to Bitfile Converter</i>	
						<i>Write Mixer & Compressor Modules</i>	
						<i>Finishing Touches</i>	
	9	10	11	12	13	14	15
<i>[Task]</i>							
<i>[Task]</i>							
<i>[Task]</i>							
<i>[Task]</i>							

Thank you...