



Phantom Sight Reader

Lance Collins, Jing Han
Dilini Warnakulasuriyarachchi

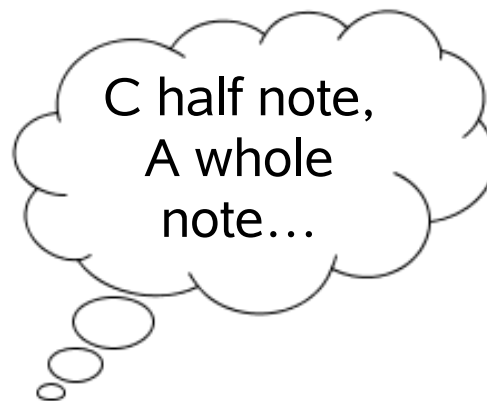
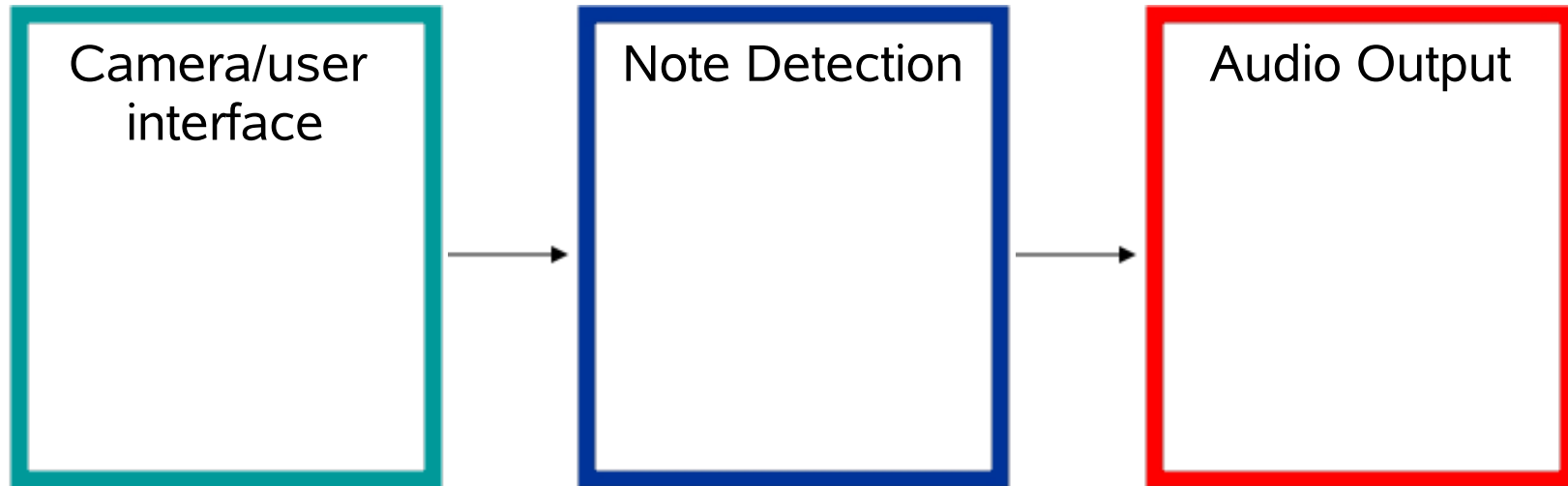
11/12/08

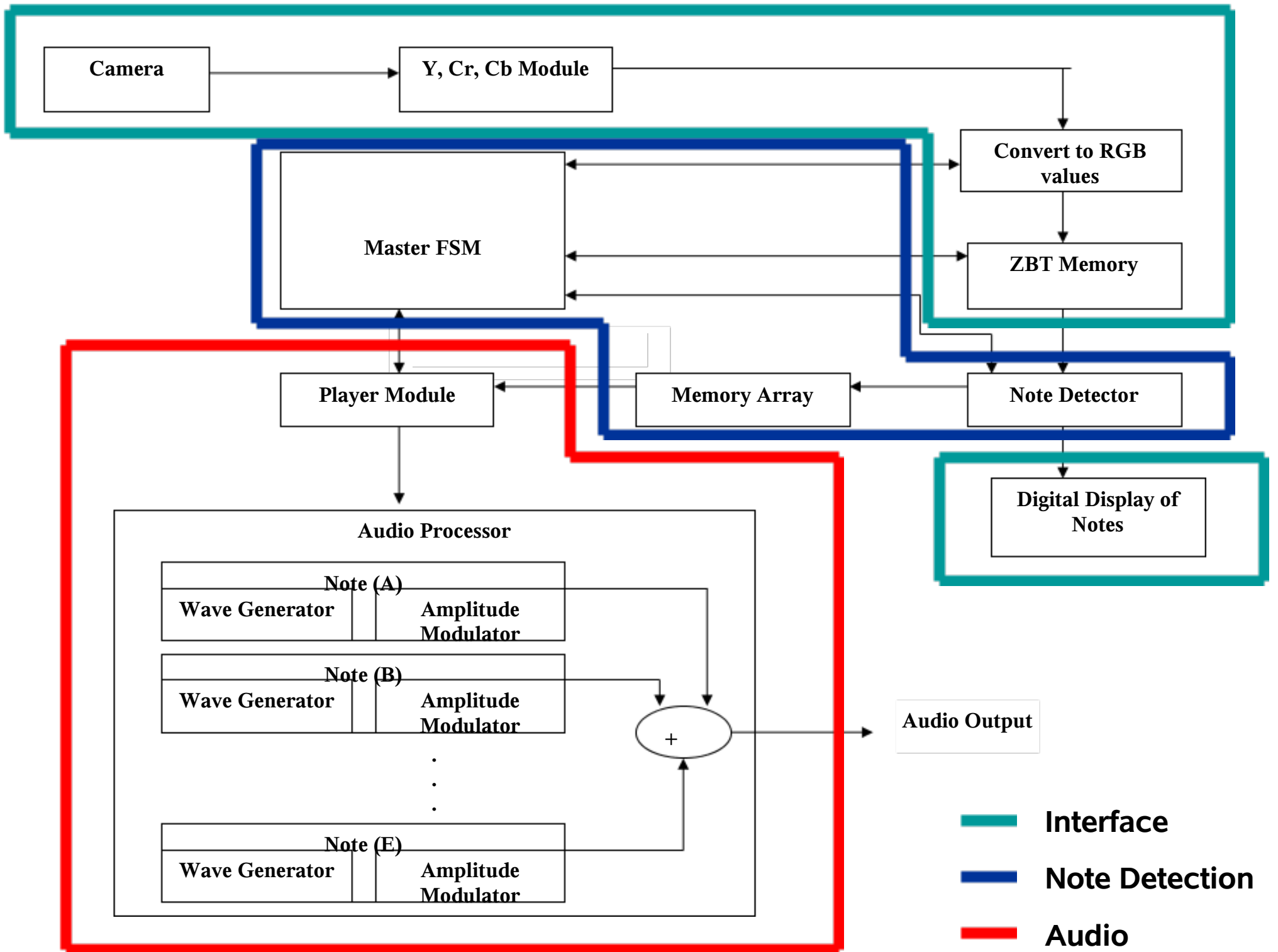
Agenda



- Overview
- Block Diagram
- Camera / User interface
- Note Decoder
- Audio Generation
- Timeline
- Conclusion

Overview





Camera/user interface - Jing

- Converting Y,Cr,Cb to RGB -> display
- Freeze image
- Filter – image B&W only, no grayscale
- Orientation of sheet music on camera
- User interface – underline note being played

Note Decoder Overview - Dilini

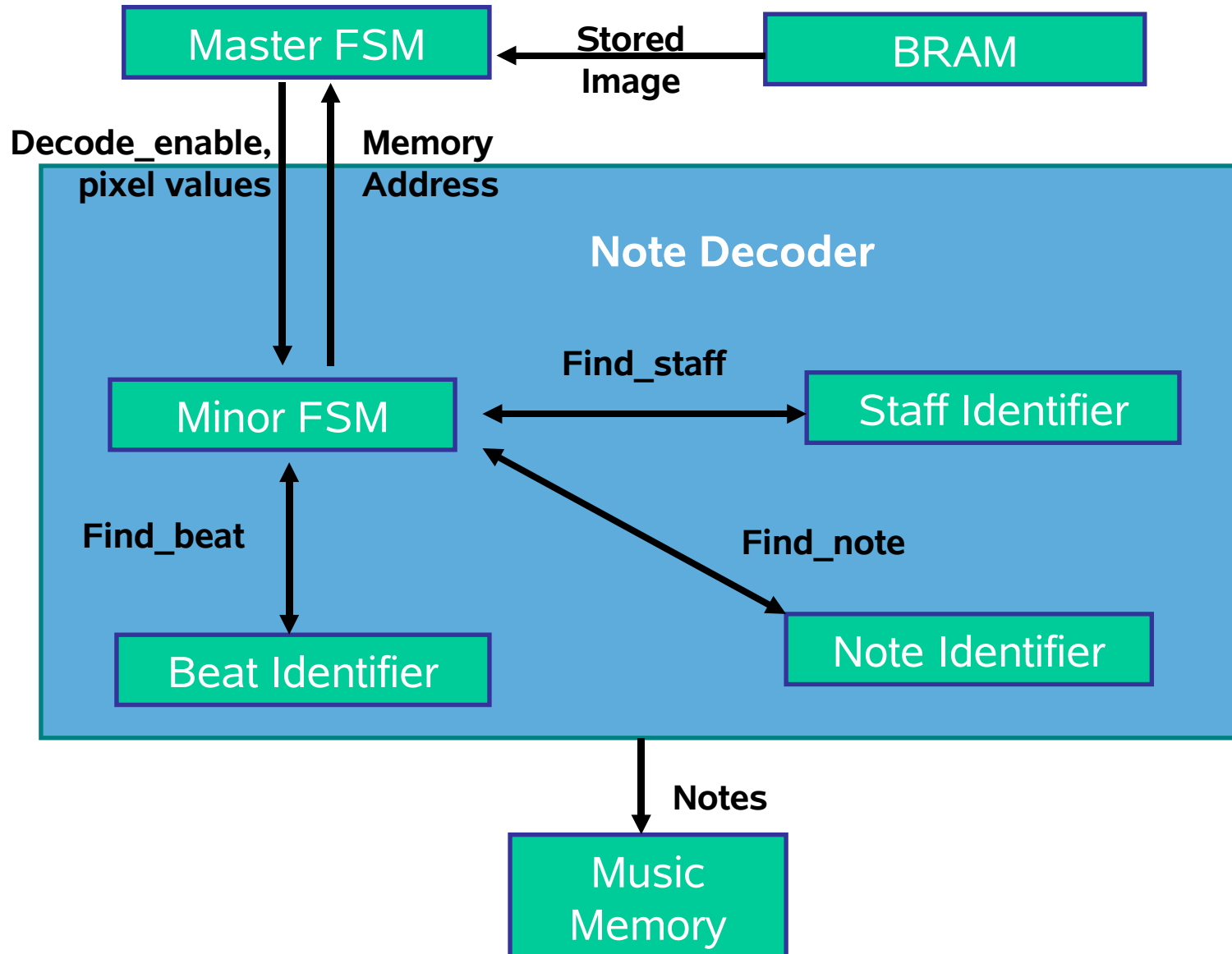
- Three main components:
 - The Staff Identifier
 - The Note Identifier
 - The Beat Identifier
- Minor FSM controls the overall function
- Interacts with the ZBT and stores data in a Memory Array to be used by the player module

Filter-pixel

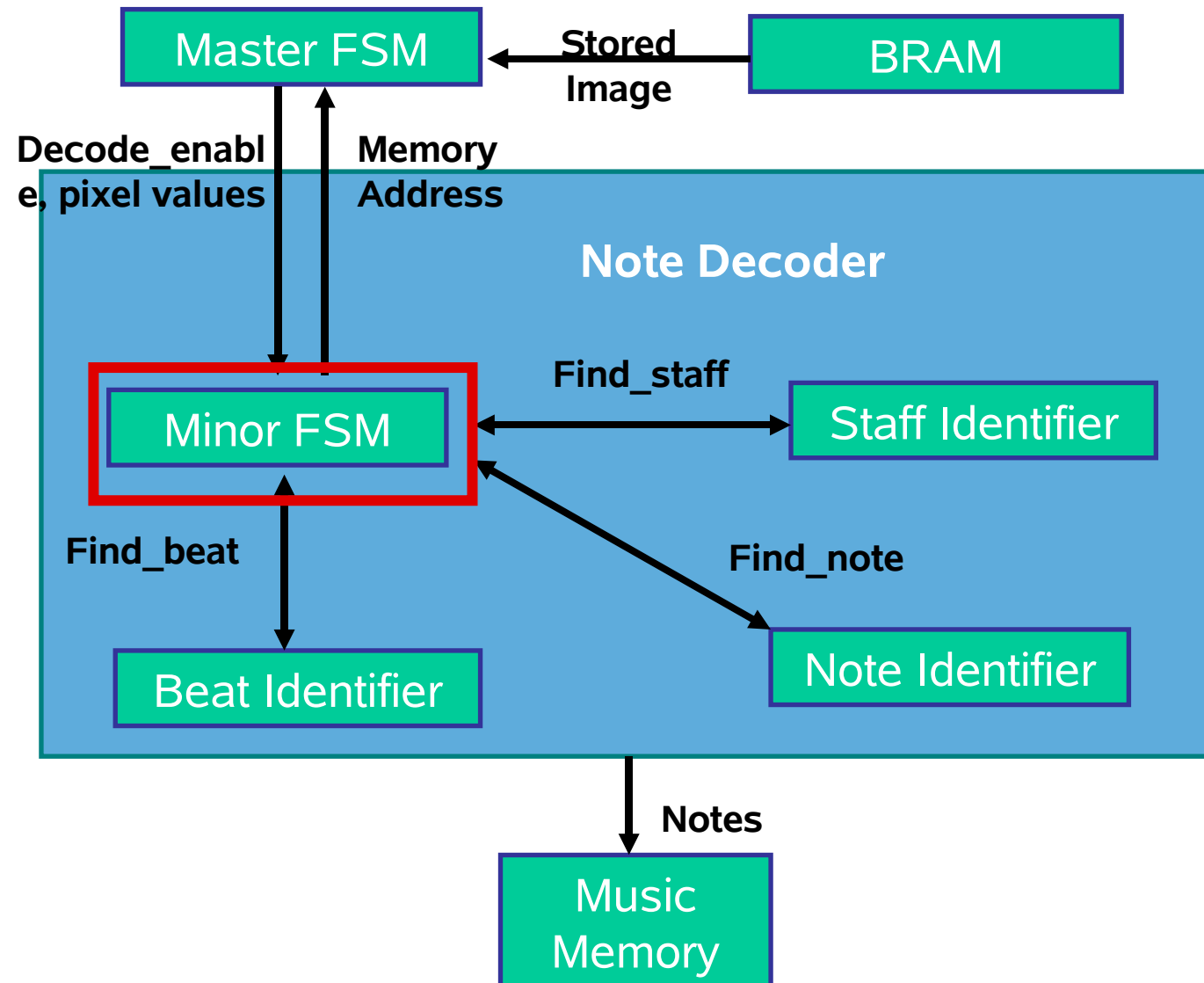
- Cleans the image captured from the camera to reduce pixel errors



Note Decoder Module

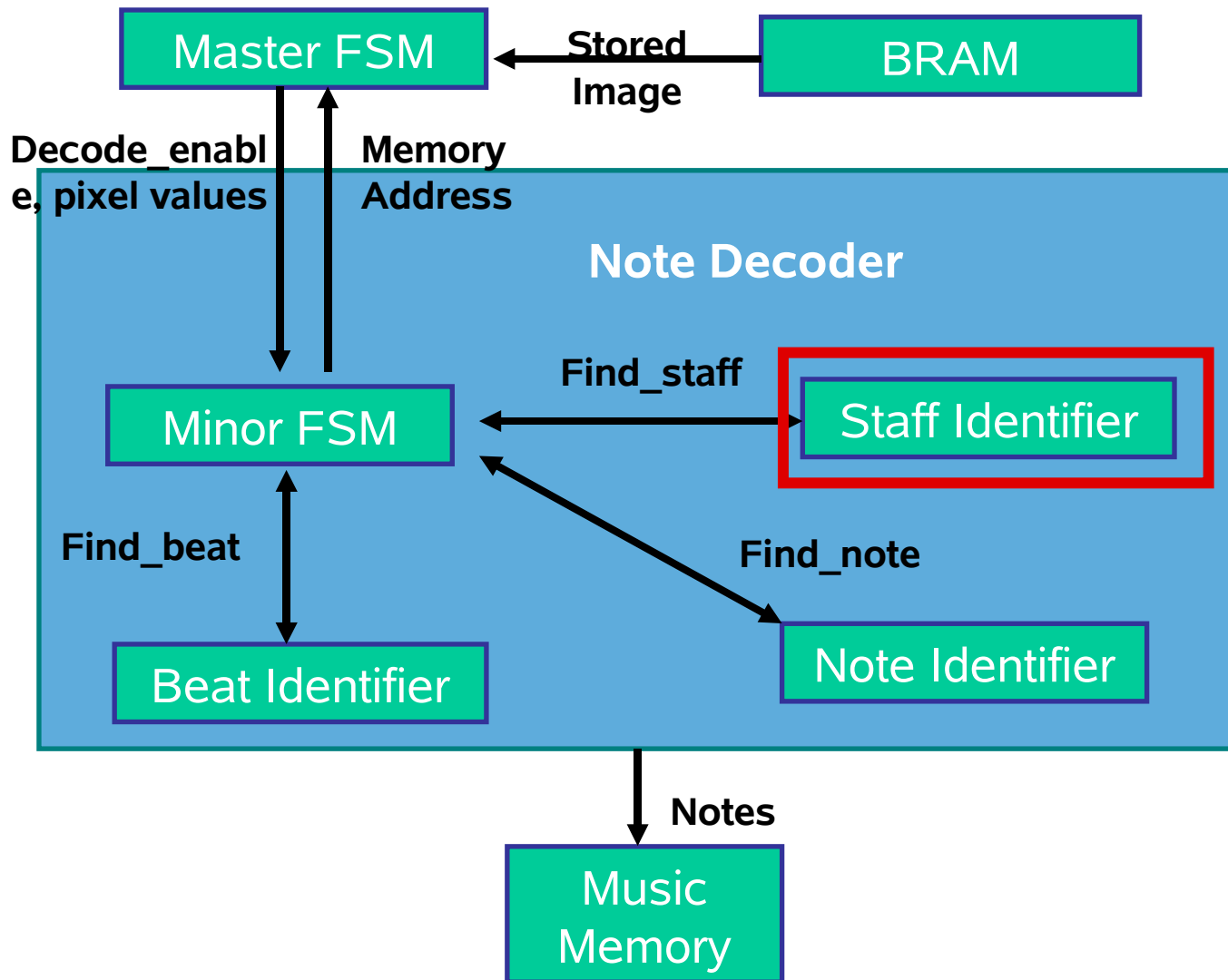


Minor Finite State Machine (FSM)



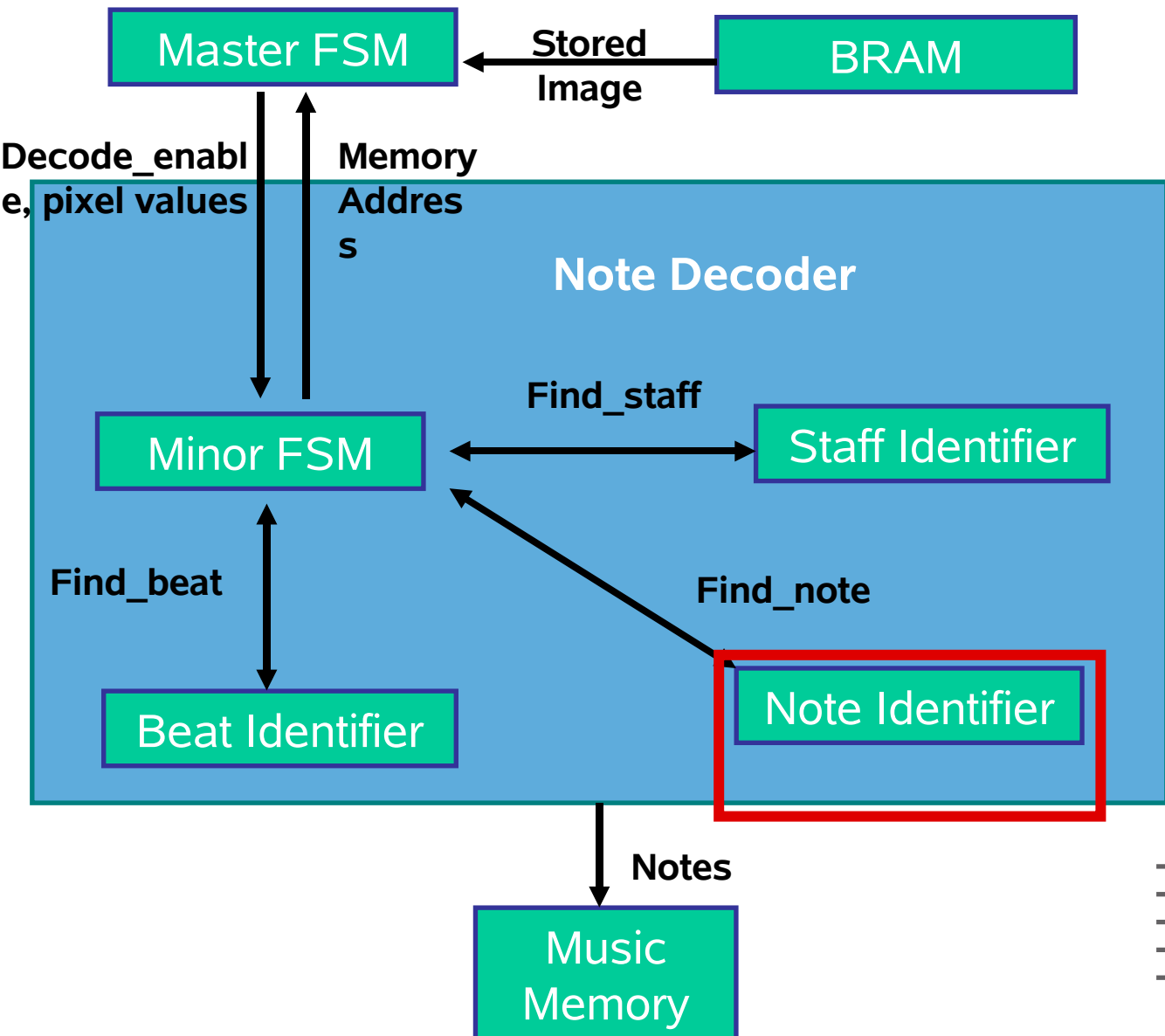
- Receives Decode_enable signal from the master FSM
- Control the sub modules in the note decoder module
- Comprised of three states
 - Staff
 - Note
 - Beat
- Request data from the BRAM via Master FSM depending on the state

Staff Identifier Module



- Locates where each staff is situated in the image
- Waits for the Find_staff enable signal from the Minor FSM
- Locates one staff at a time and notes the location
- Once a staff is found notifies the Minor FSM by setting Find_staff signal to low

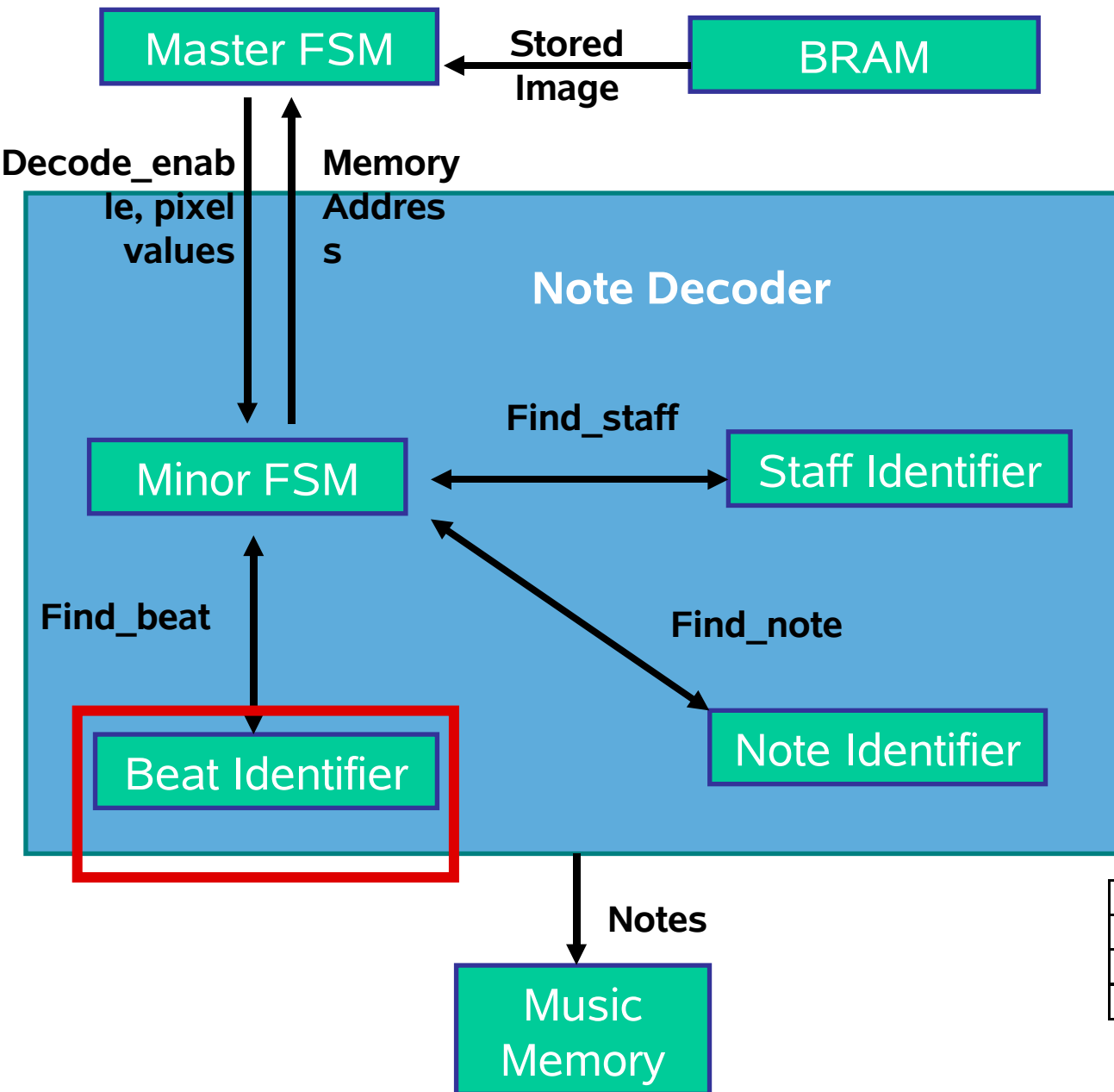
Note Identifier Module



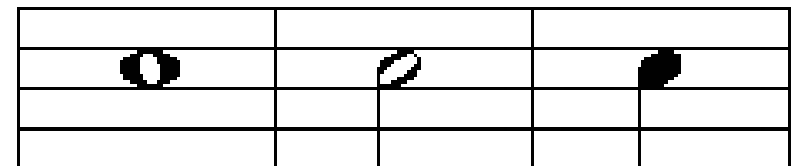
- Identifies each note by evaluating the number of black pixels in each block between staff lines
- Activated by the Find_note signal from the Minor FSM
- The notes on the score sheet must be evenly spaced



Beat Identifier



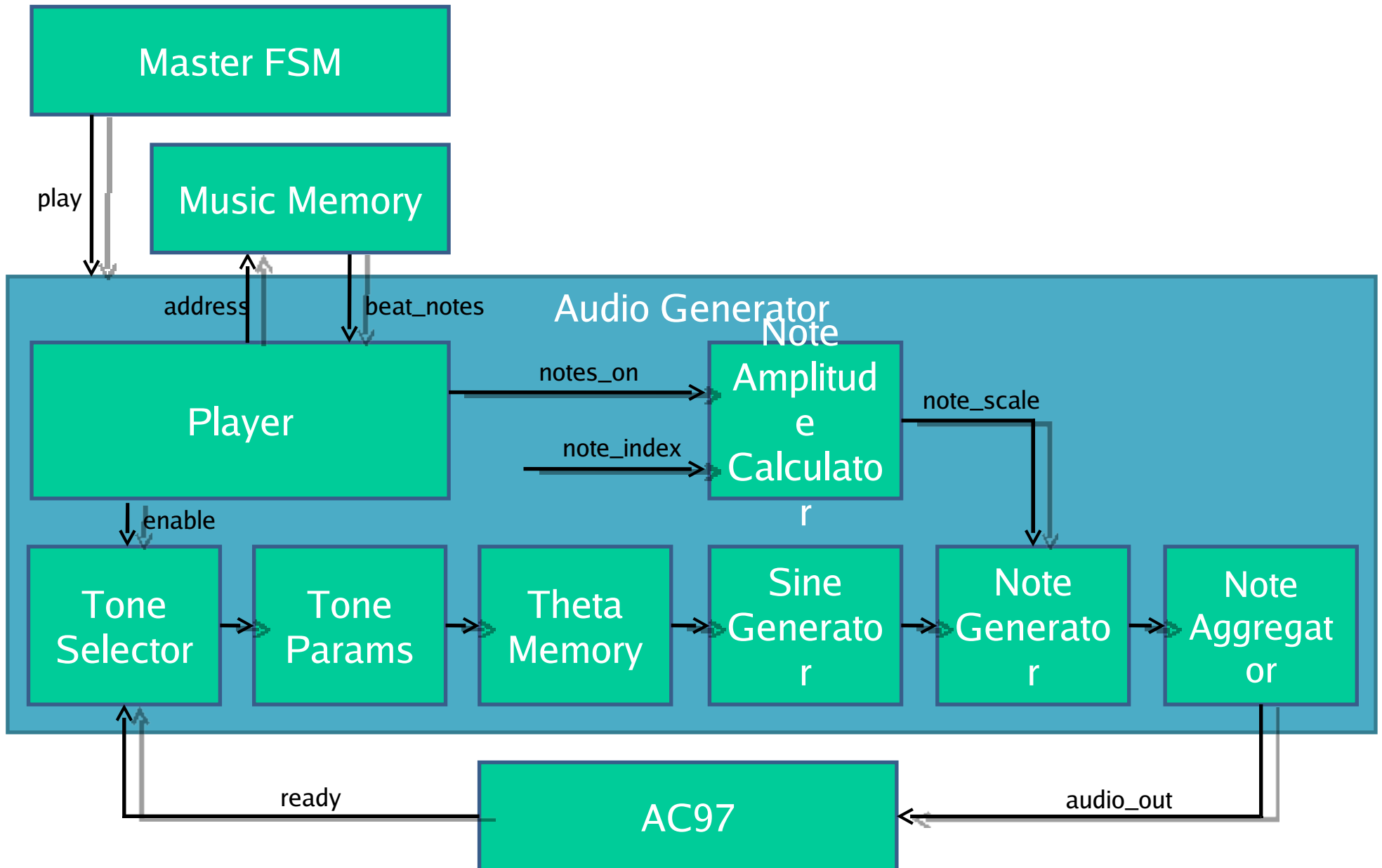
- Identify the duration of each note
- Waits for the Find_beat signal from the Minor FSM
- Counts the number of black pixels in each column of a single staff



Audio Generation Overview - Lance

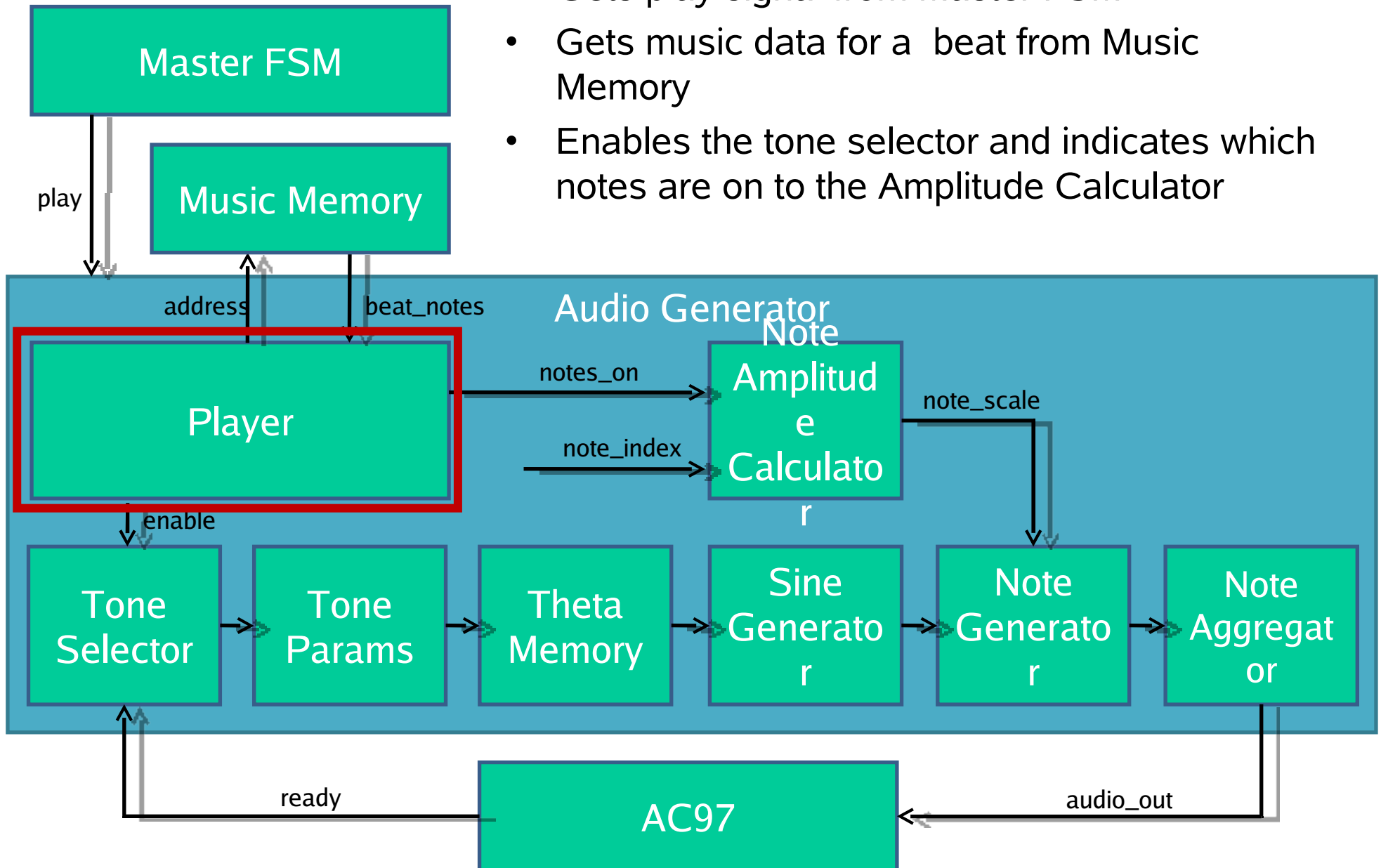
- Initialization
 - Music Memory filled with music data
 - Play signal from Master FSM
- Operation
 - Ready pulse from AC97
 - Begin computing note data
 - Pipelined design
 - On next ready pulse, output audio data
 - Compute note tones (amplitude and phase)
 - Apply Attack-Delay-Sustain-Release (ADSR) envelope
 - Combine notes together

Audio Generator Module

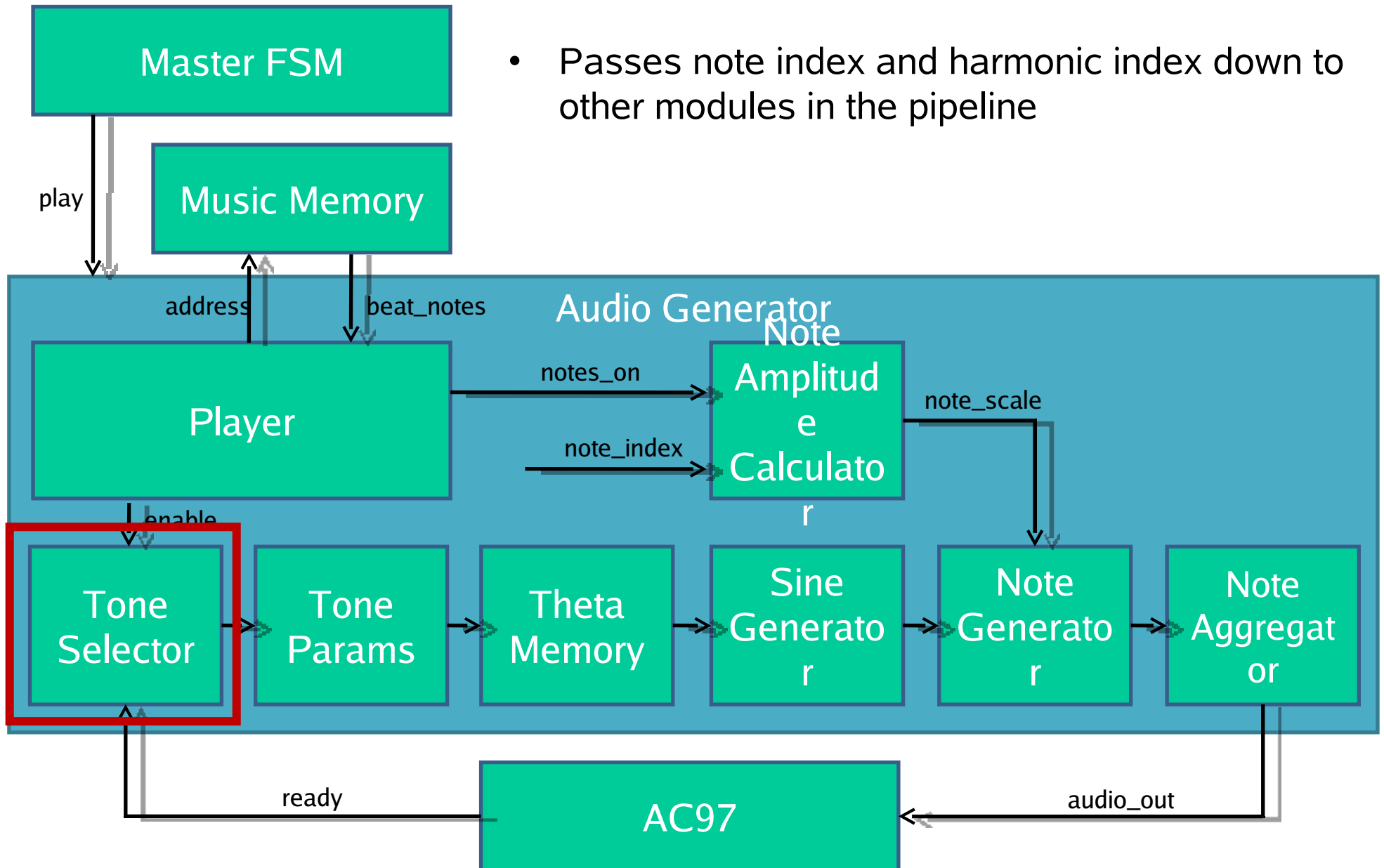


Player Module

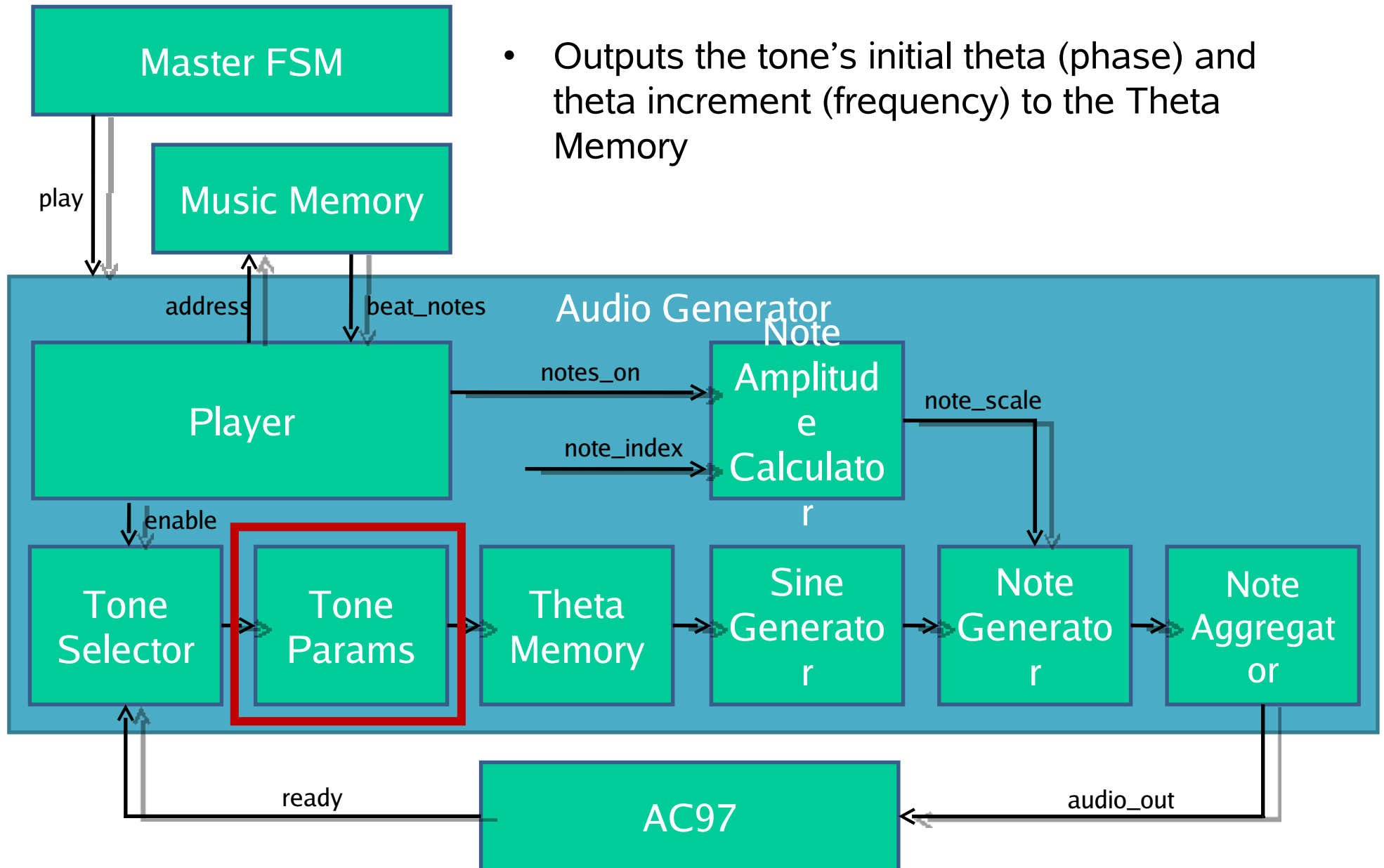
- Gets play signal from Master FSM
- Gets music data for a beat from Music Memory
- Enables the tone selector and indicates which notes are on to the Amplitude Calculator



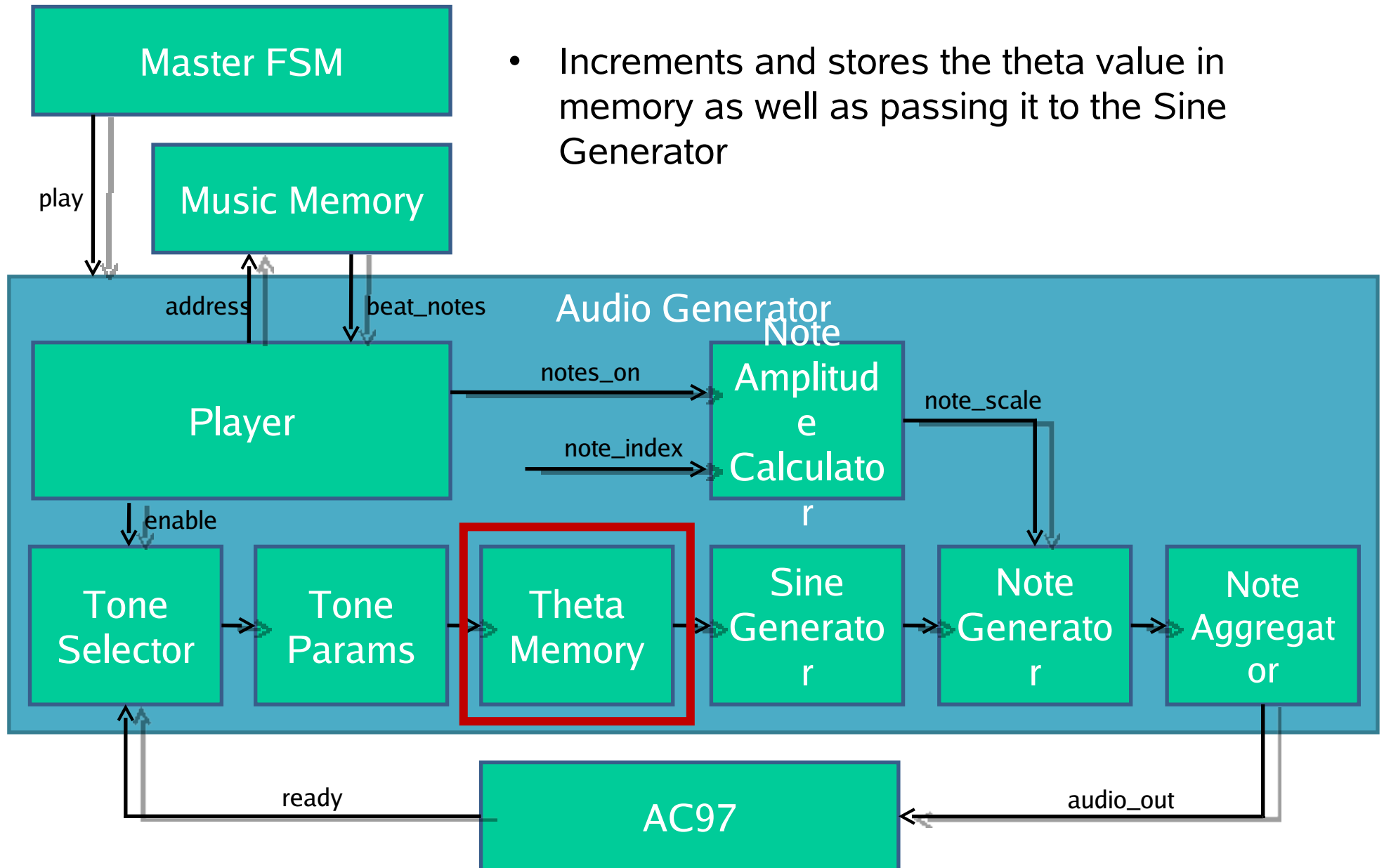
Tone Selector Module



Tone Parameters Module

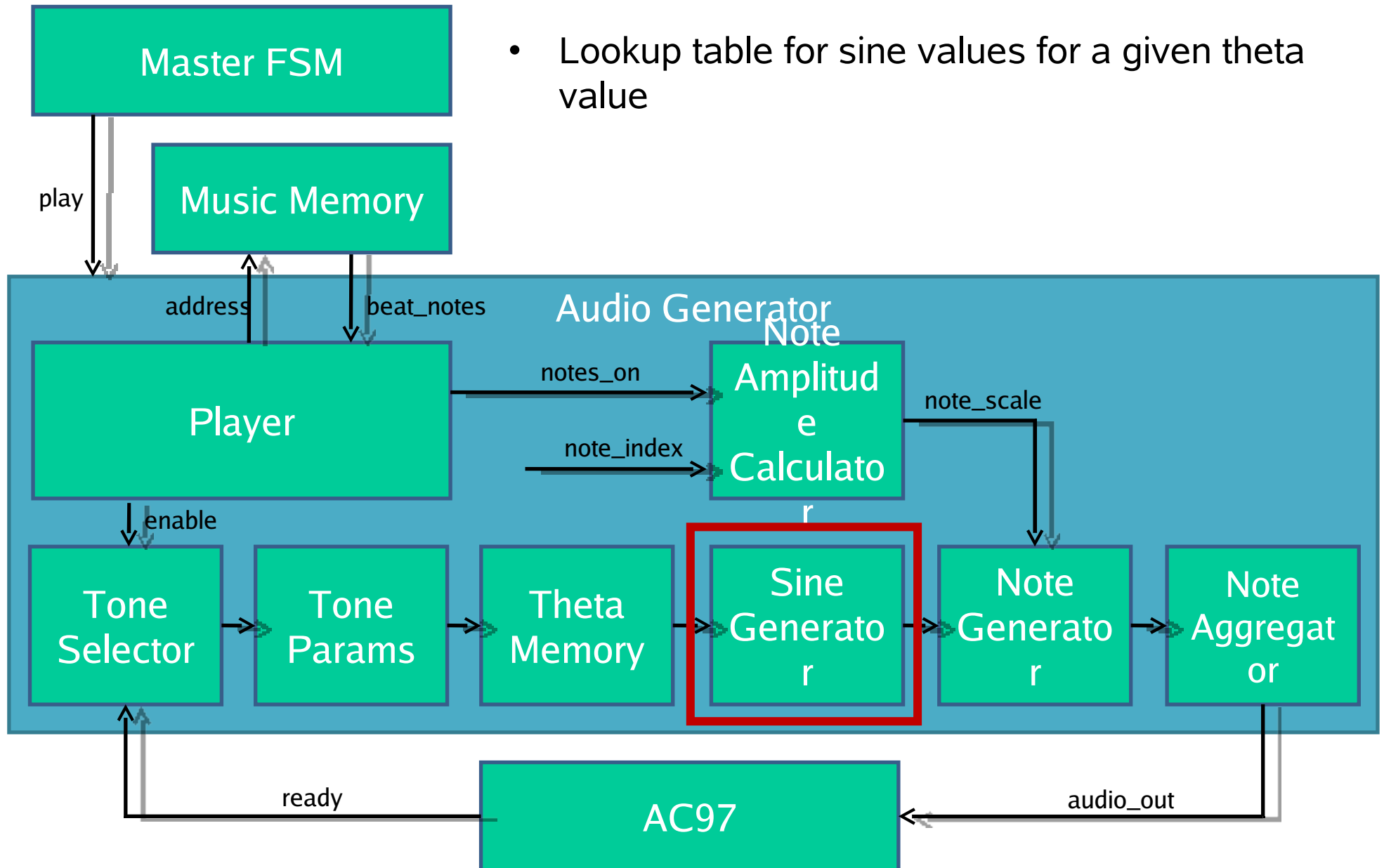


Theta Memory Module



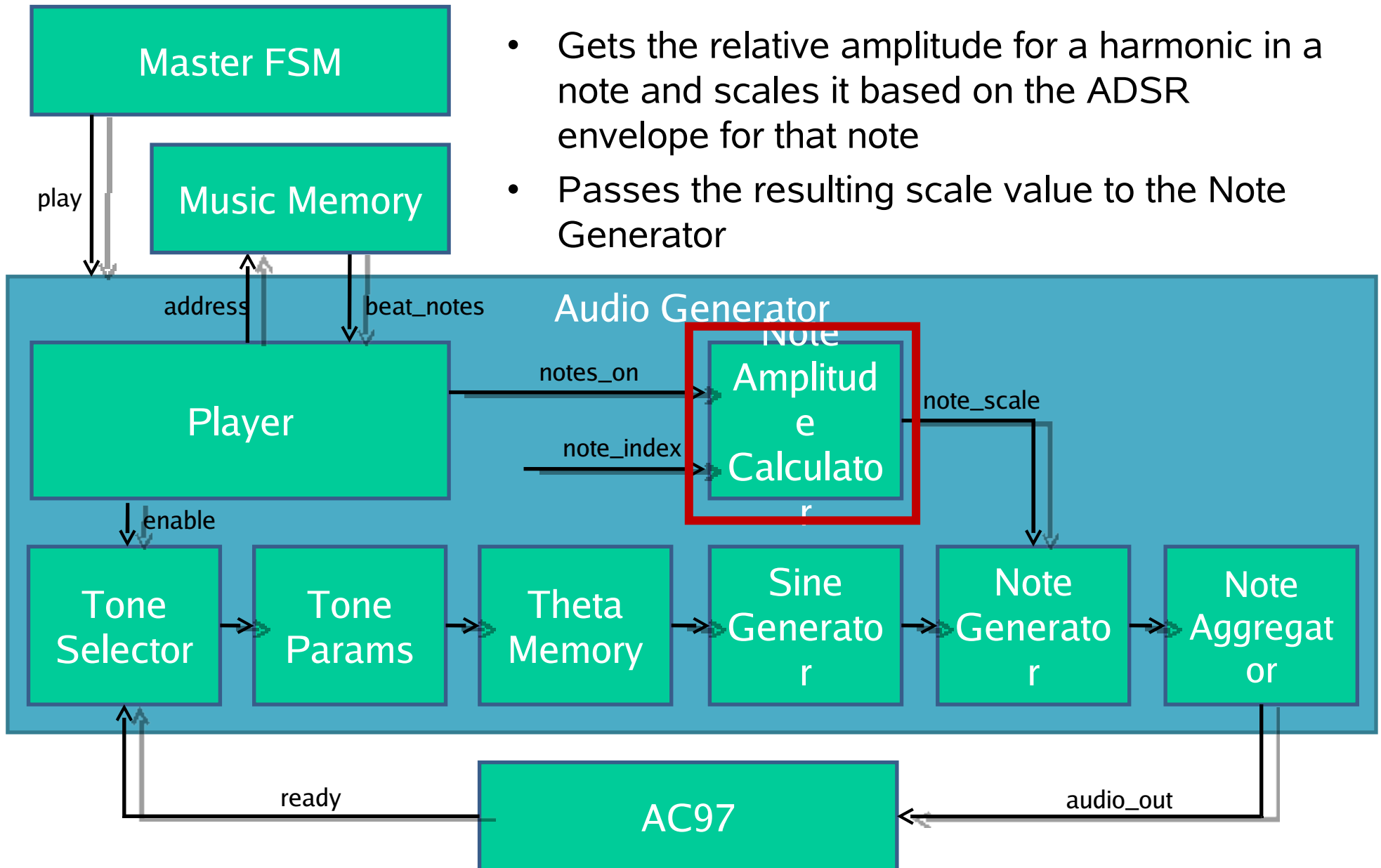
- Increments and stores the theta value in memory as well as passing it to the Sine Generator

Sine Generator Module



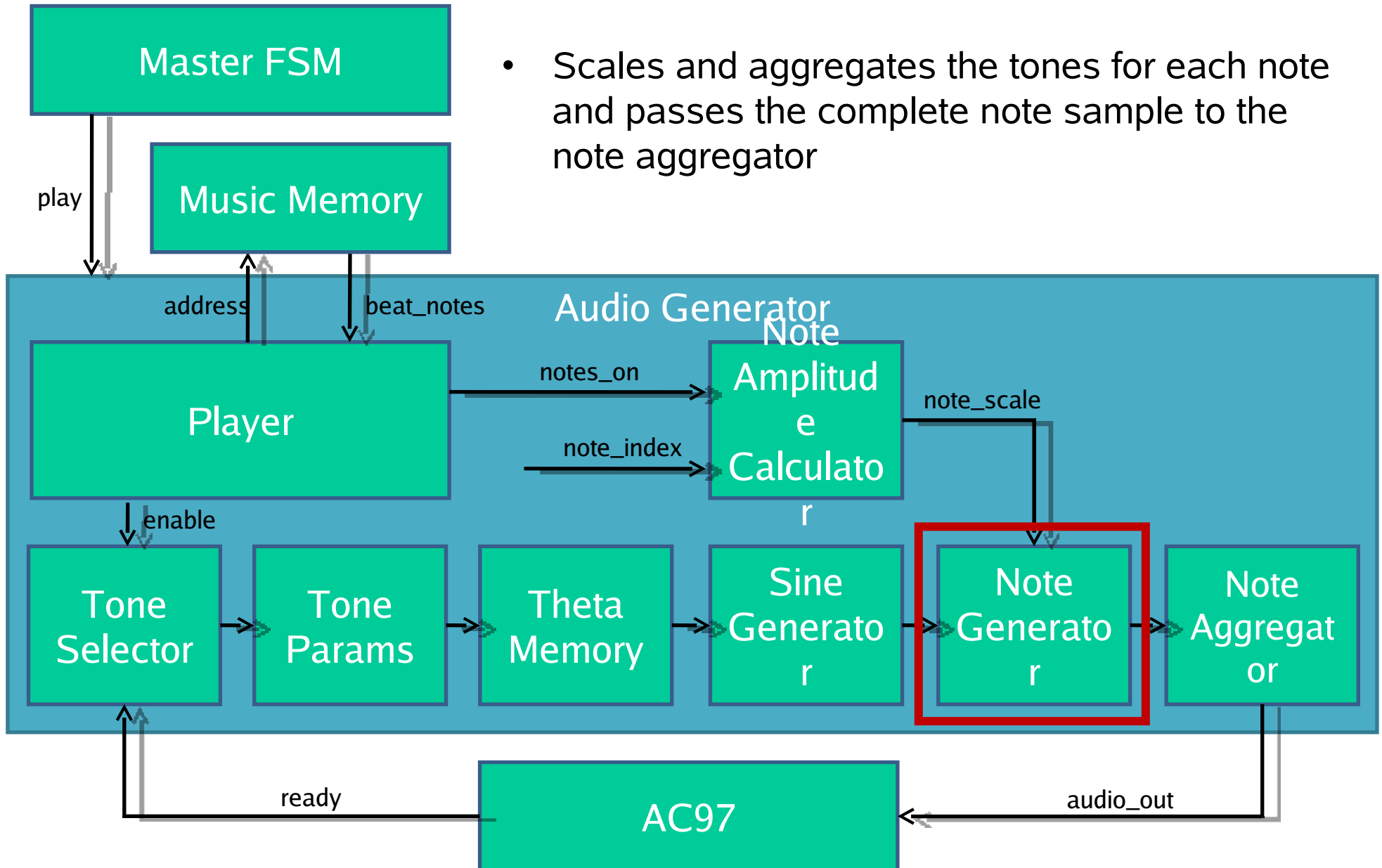
- Lookup table for sine values for a given theta value

Note Amplitude Module

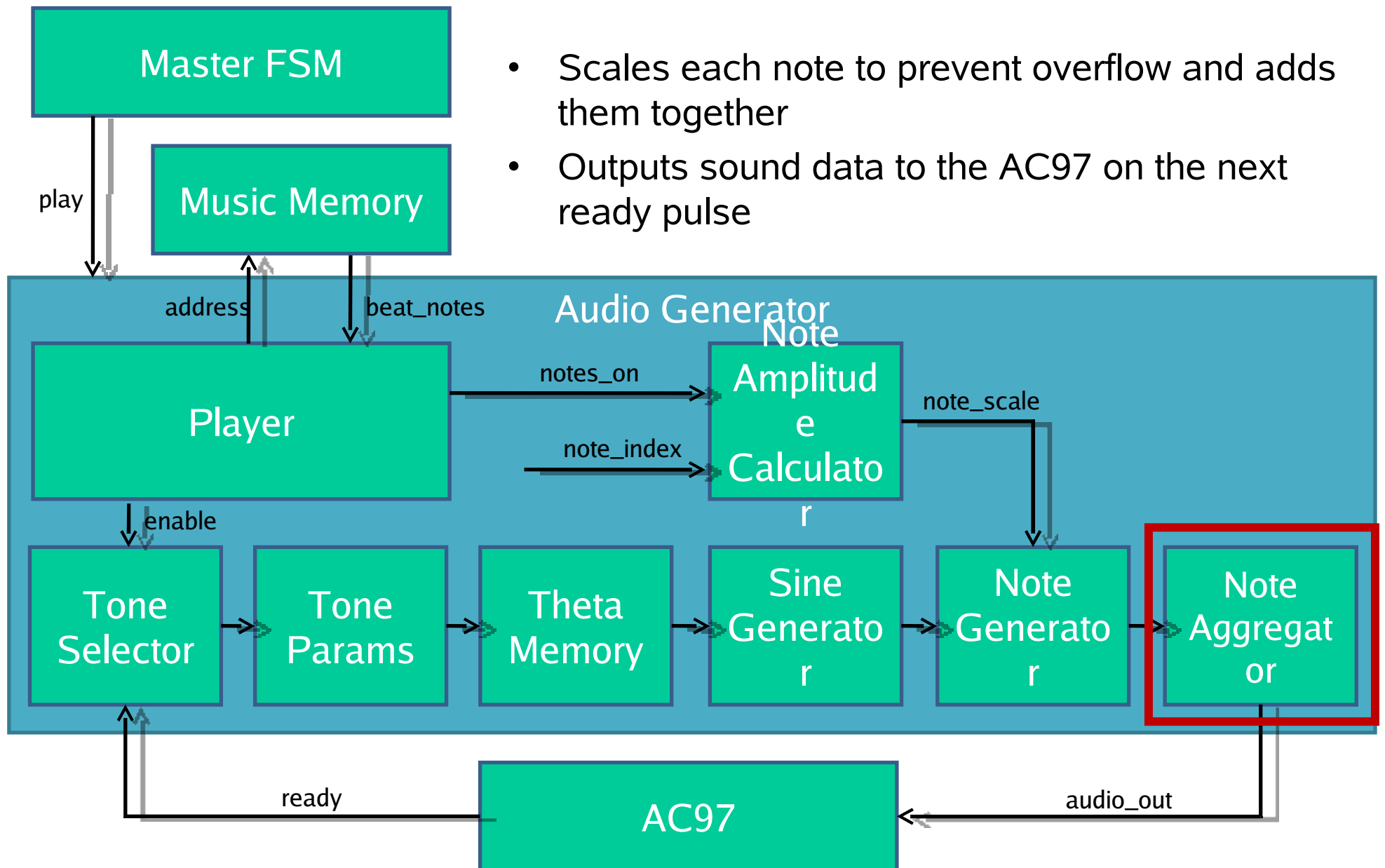


- Gets the relative amplitude for a harmonic in a note and scales it based on the ADSR envelope for that note
- Passes the resulting scale value to the Note Generator

Note Generator Module



Note Aggregator Module



- Scales each note to prevent overflow and adds them together
- Outputs sound data to the AC97 on the next ready pulse

Timeline

	Dilini	Jing	Lance
Week 1	<ol style="list-style-type: none">1. Basic Pixel Concentration Note Recognition2. Basic Pattern Matching Note Recognition	<ol style="list-style-type: none">1. Filter Module2. Orientation Module	<ol style="list-style-type: none">1. Multiple Notes2. Player Module3. Apply ADSR envelopes
Week 2	<ol style="list-style-type: none">1. ZBT interface2. Choose note detection methodology and expand	<ol style="list-style-type: none">1. Underline notes as played2. UI Elements	<ol style="list-style-type: none">1. Extend to Multiple Instruments
Week 3	Integration		
Week 4	Debugging		

Conclusion

- Baseline
 - Reads a few staves of preformatted sheet music
 - Plays the sheet music using piano or other
 - Shows notes being played
- Optimal
 - Reads a pages of preformatted sheet music
 - Users customizes which instruments play various parts (UI and Audio components)