Team 10

Matthew Arceri

Vasundhara Rawat

Will Sutton

Philip Wolfe

FPGA Clapping Pattern Detector

Preliminary Design Review

Goal:

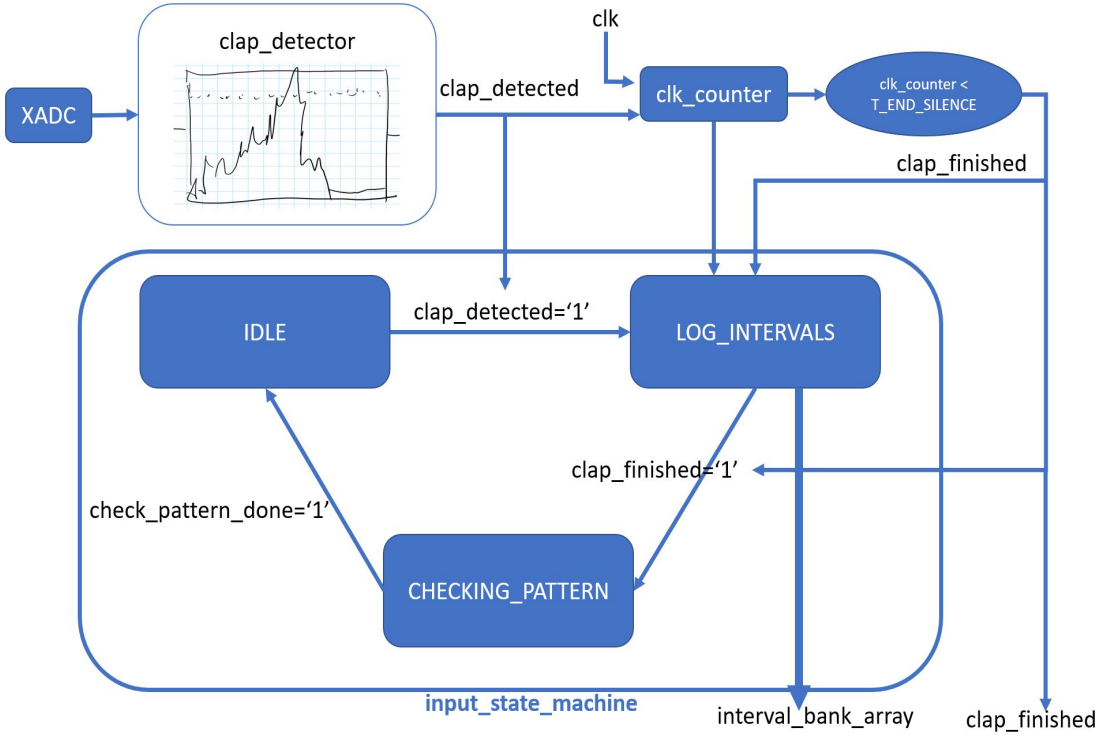
Detect and reject certain clapping patterns and provide visual feedback.

Outline

- Project Block Diagram and Individual Component Description (2 slides)
- Planned optimizations for any sub-systems (1 slides)
- Timeline for the Project and each team member’s work (1 slide)
- Team member work distribution (1 slide)
- Risks and Plans taken to mitigate them (1 slide)
- Prior Work and Literature Review. cite full references (3 slides)
- Expected results (1 slides)

Project Block Diagram Individual Component Description

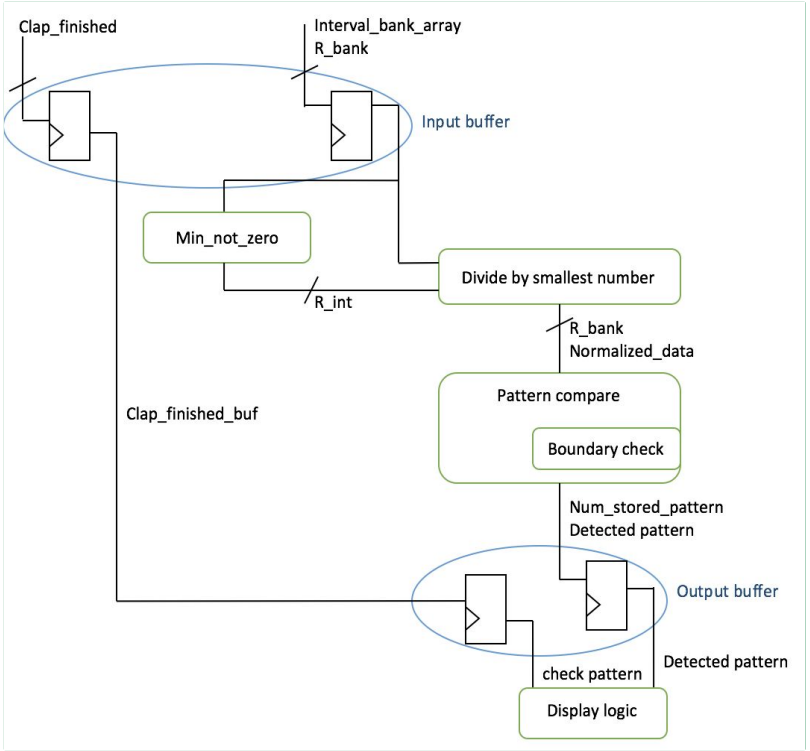
- **XADC -** Xilinx ADC to sample the microphone
- **clap_detector -** takes samples, filters, and detects a clap
- **clk_counter** - counts clock cycles to log intervals between claps and detect end. Reset when clap is detected
- **input_state_machine** - manages logging and flushing *interval_bank_array* (buffer for intervals between claps)
- **check_valid_pattern** - while in *CHECKING_PATTERN* it checks if recorded pattern matches a stored valid pattern



Project Block Diagram Individual Component Description Continued

check_valid_pattern:

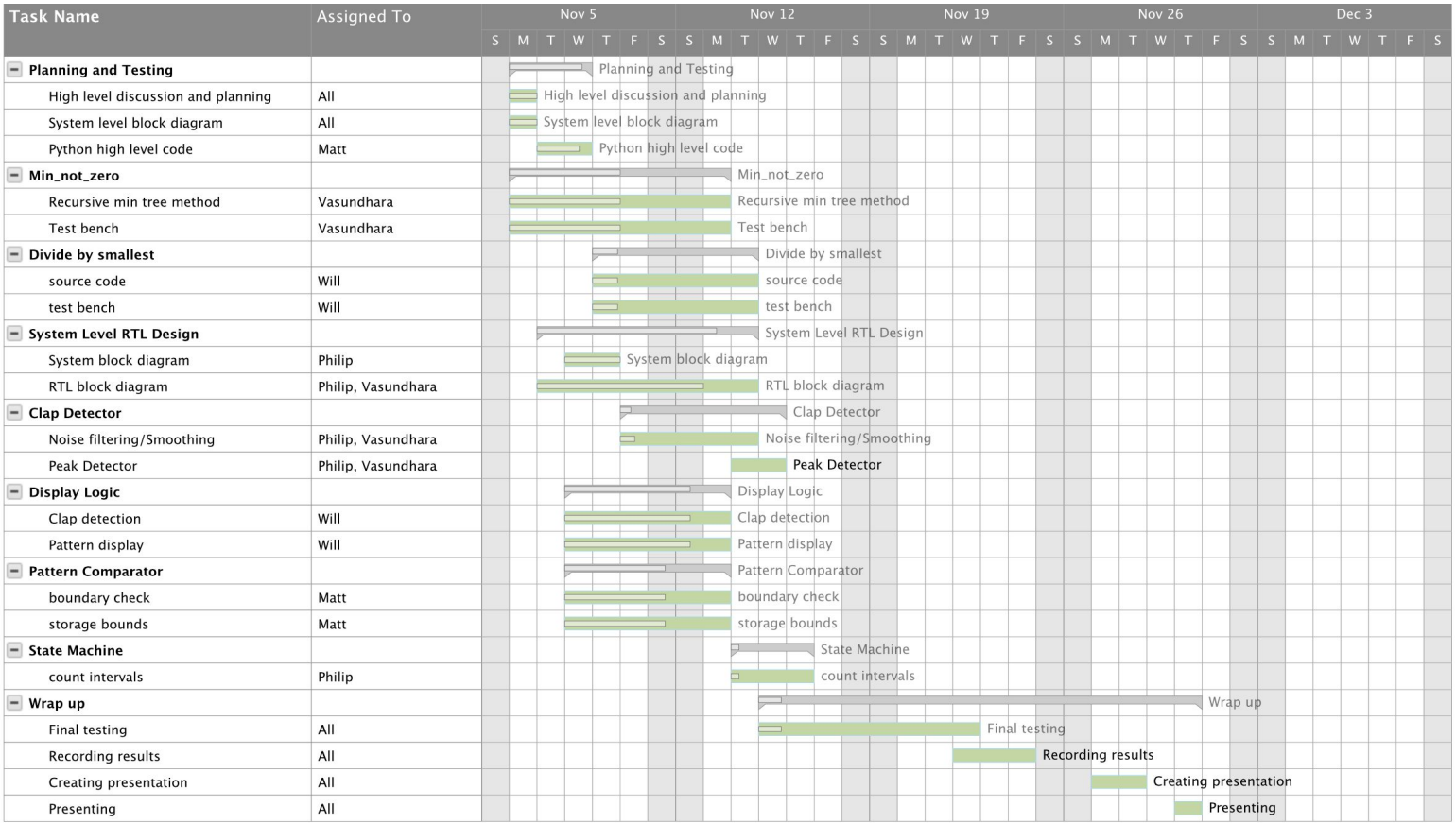
- **min_not_zero** - takes *interval_bank_array* and returns the smallest non-zero value
- **Divide_by_smallest_number** - divides *interval_bank_array* by smallest value from *min_not_value*
- **Pattern_compare** - compares to see which patterns match using *Boundary_check*. If all intervals fall within the boundaries for a pattern, then the pattern is detected
- **Boundary_check** - checks an interval to see if it falls within the boundary of a stored pattern interval
- **Display logic** - controls seven segment display visual feedback based on current state and if a pattern has been detected/rejected



Planned Potential Optimizations

- RTL optimization
  - Preloading clocks (RTL optimization)
  - Flip flop clock frequency divider
- Optimizing sampling frequency (choosing freq to have enough samples to properly detect a clap)
- Area vs Speed optimizations (computing matching boundaries sequentially instead of concurrently/etc)
  - Finding min via tree vs reusing one comparing module over time
- Algorithmic optimizations:
  - Investigate more efficient ways of computing the boundary comparisons

Project Timeline



	Task Name	Assigned To
1	**Planning and Testing**	
2	High level discussion and planning	All
3	System level block diagram	All
4	Python high level code	Matt
5	**Min_not_zero**	
6	Recursive min tree method	Vasundhara
7	Test bench	Vasundhara
8	**Divide by smallest**	
9	source code	Will
10	test bench	Will
11	**System Level RTL Design**	
12	System block diagram	Philip
13	RTL block diagram	Philip, Vasundhara
14	**Clap Detector**	
15	Noise filtering/Smoothing	Philip, Vasundhara
16	Peak Detector	Philip, Vasundhara
17	**Display Logic**	
18	Clap detection	Will
19	Pattern display	Will
20	**Pattern Comparator**	
21	boundary check	Matt
22	storage bounds	Matt
23	**State Machine**	
24	count intervals	Philip
25	**Wrap up**	
26	Final testing	All
27	Recording results	All
28	Creating presentation	All
29	Presenting	All

Team Member Work Distribution

- Will Sutton: Clock divider and 7-Segment display management, data normalization
- Vasundhara Rawat: min-not-zero, clap detection(noise filtering, smoothing, peak detection)
- Philip Wolfe: general project management, system level design, rtl block diagrams, interval bank state machine, assist VHDL for clap detection
- Matthew Arceri: Pattern Comparator, Pattern Storage, Boundary Checker Modules

Risks and Risk Mitigation

Risk: Microphone audio quality too low
Severity: Medium
Likelihood: Medium
Mitigation: Acquire a lower noise/more sensitive microphone or reduce the distance used for real-world testing.

Risk: Humans are unable to consistently perform patterns
Severity: Medium
Likelihood: Low
Mitigation: Create an adjustable boundary “knob” that loosens or tightens the boundaries as needed.

Prior Work / Literature Review

https://forums.ni.com/t5/LabVIEW/How-to-make-a-clapping-detection-program/td-p/1696160

- National Instruments Forum: How to detect clapping patterns
  - Log intervals between claps with shift register
  - Compare values to threshold for pattern detection

Prior Work / Literature Review Continued

https://github.com/iver56/clap-detection

- Uses slope detection to detect a clap.
- Compares time of each clap against preset patterns and has a 10% allowable variance in timing.
- Normalizes data before analyzing

Prior Work / Literature Review Continued

http://ieeexplore.ieee.org/document/4147154/

- Genetic Algorithm based feature extraction for detecting a clapping sound.
- Not clearest method.
- Highest and lowest MFCC’s are more discriminative for clap sounds than middle

Expected Results:

- Finished single clap detector
- Finished state machine to store raw intervals between clap and flush each time a new pattern comes in
- Finished check_valid_pattern combinational logic block & testbench
- Finished display logic
- Basys3 programed to successfully detect/reject clapping patterns

Questions?