VOICE-CONTROLLED CHESS GAME ON FGPA USING DYNAMIC TIME WARPING

Michael Kuo

Varun Chirravuri

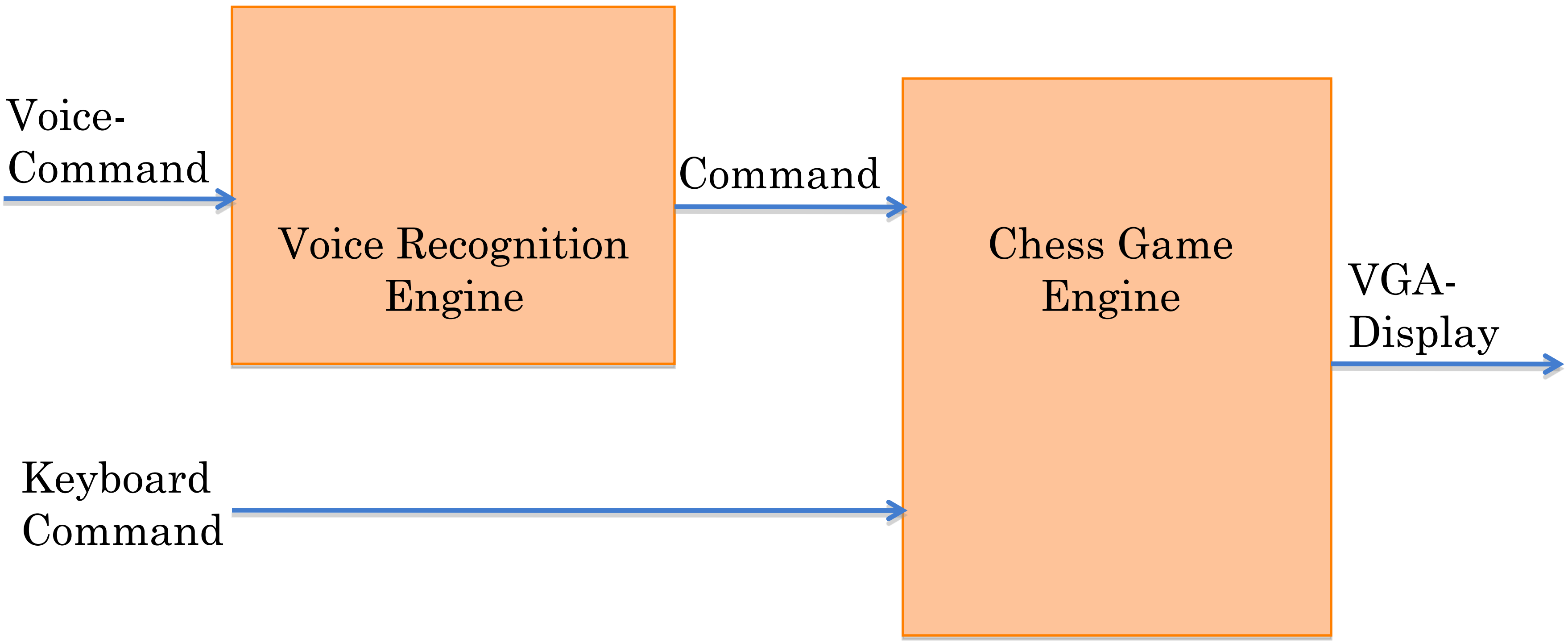
OVERVIEW

- Multiplayer digital chess game
- Players deliver voice commands to game
- Driven by voice recognition hardware and move-checking logic

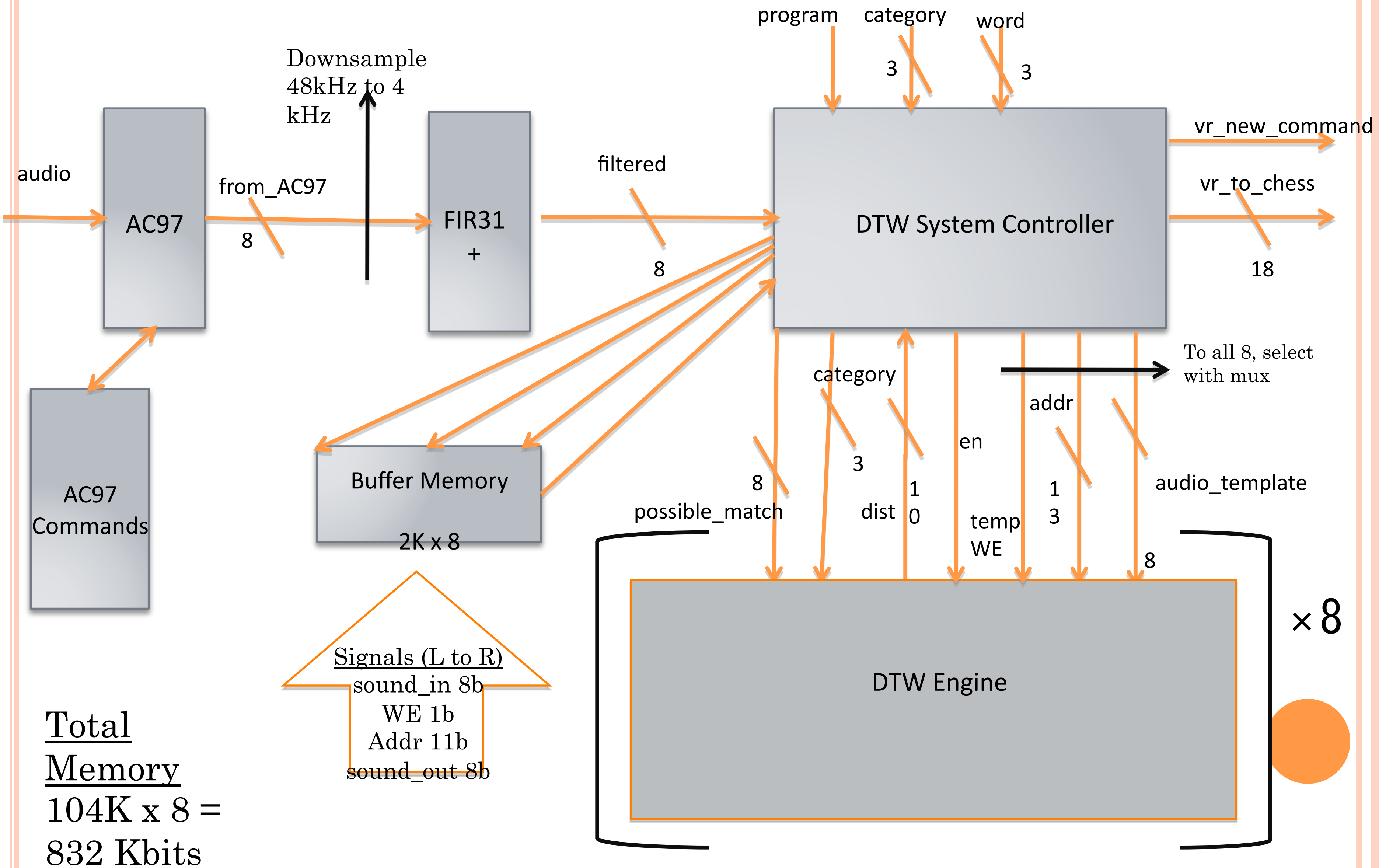
DIVISION OF LABOR

- The project is divided into two parts:
- 1. Voice Recognition Engine (Varun)
  - Compares voice commands to saved samples
  - Uses Dynamic Time Warping (DTW) Algorithm to compare
  - Returns command corresponding to closest match
- 2. Chess Game Engine (Michael)
  - Takes input from the Voice Recognition Engine or keyboard
  - Checks the validity of moves
  - Displays the chessboard and chess pieces on an XVGA display

HIGH LEVEL BLOCK DIAGRAM



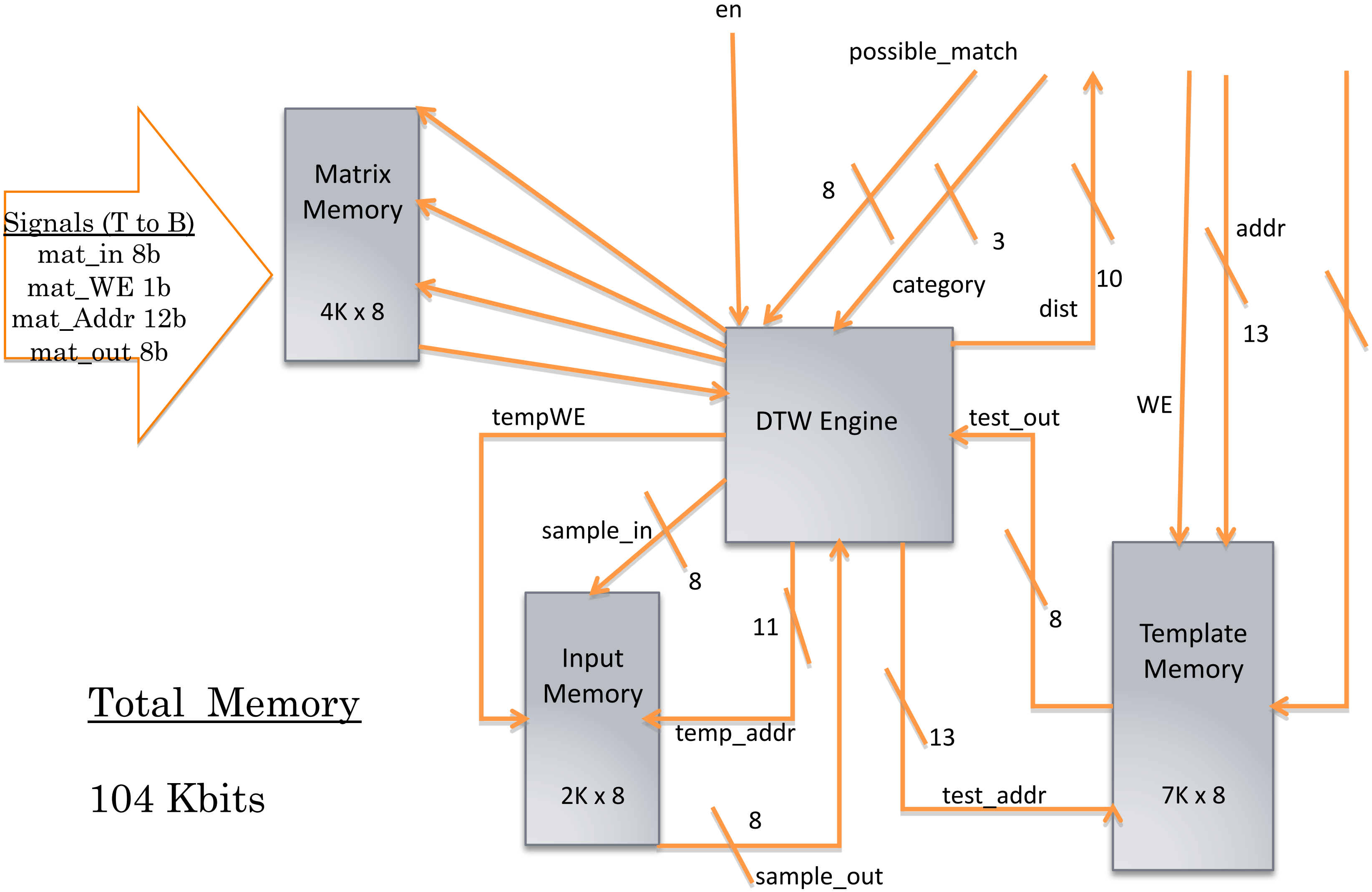
VOICE RECOGNITION ENGINE



Signals (L to R)
sound_in 8b
WE 1b
Addr 11b
sound_out 8b

Total Memory
104K x 8 =
832 Kbits

DYNAMIC TIME WARPING (DTW) ENGINE



Signals (T to B)
mat_in 8b
mat_WE 1b
mat_Addr 12b
mat_out 8b

Total Memory

104 Kbits

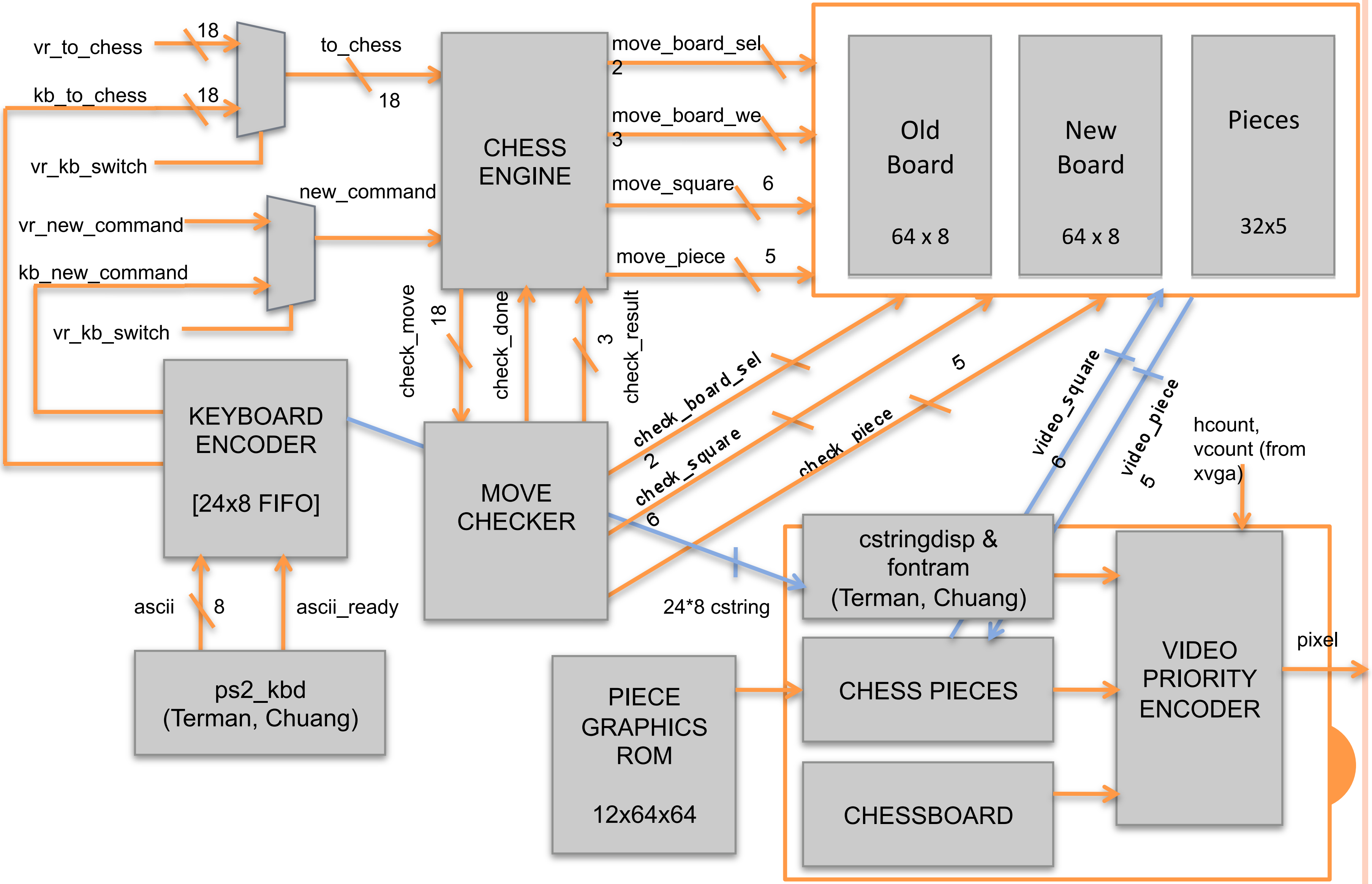
DYNAMIC TIME WARPING EXPLAINED

- Compares input sample of A (N samples) to template B (M samples) and returns a least cost “distance” between the two
  - Forms M x N matrix D, $D_{i,j} = (A_i – B_j)^2$
  - Forms second M x N matrix Gamma:
    - Row 0 and Column 0 of Gamma set to Infinity; $\text{Gamma}_{0,0} = 0$
    - For all other elements $\text{Gamma}_{i,j} = D_{i,j} + \min(\text{Gamma}_{(i-1,\ j)}, \text{Gamma}_{(i-,\ j-1)}, \text{Gamma}_{(i-1,\ j-1)})$

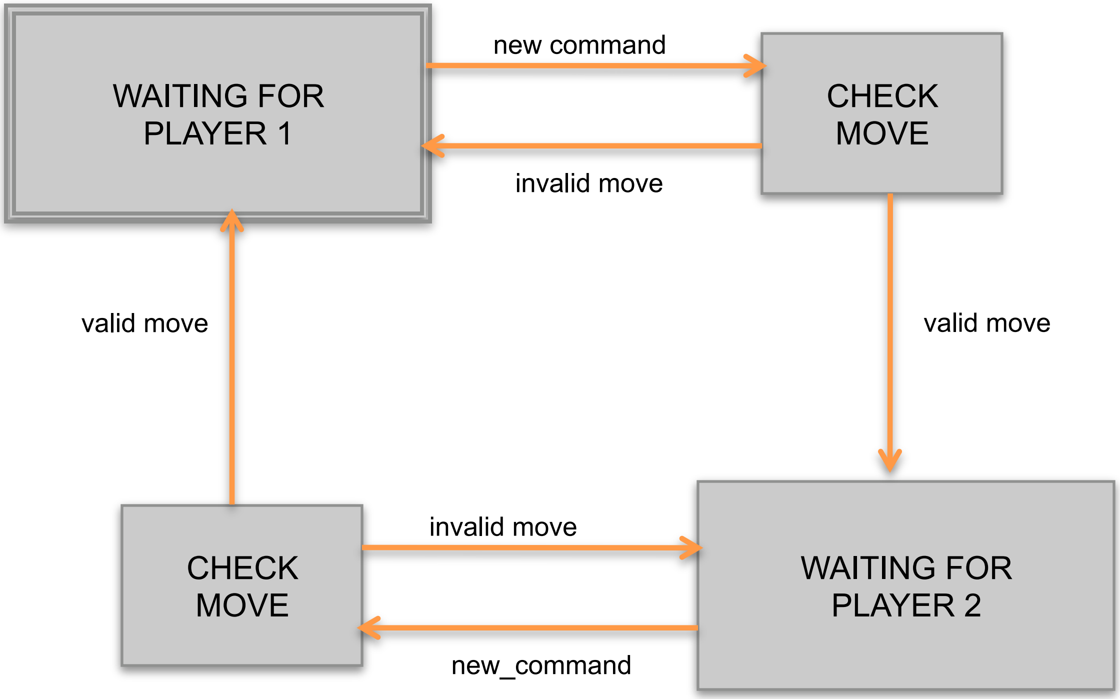
$\text{Gamma}_{(i-,\ j-1)}$	$\text{Gamma}_{(i-1,\ j)}$
$\text{Gamma}_{(i-1,\ j-1)}$	$\text{Gamma}_{i,j}$

  - Return $\text{Gamma}_{N,\ M}$ as the distance measure between A and B
- Optimizations
  - Our maximum N and M are 2K for a memory requirement of 16K x 16K for both Gamma AND D
  - But…
  - We can certainly calculate D values on the fly with a pipelined circuit
  - And…
  - The algorithm only ever calls on data from two complete rows, so we only need to store 2 in a rolling buffer style memory, cutting memory down to 16K x 2

CHESS & GRAPHICS ENGINES



CHESS ENGINE FSM


WAITING FOR PLAYER 1

new command

CHECK MOVE

invalid move

valid move

valid move

CHECK MOVE

invalid move

WAITING FOR PLAYER 2

new_command

MOVE CHECKING

- Checks that proposed move is characteristic of piece's style of movement (e.g. knight from G1 to F3)
- Checks whether castling is still permissible
- Checks whether pawn can still perform a two-square advance

Extensions

- Chess Module
  - Check and Check-Mate Checking
  - Non-Queen Promotion
  - En Passant Captures
- Voice Recognition Module
  - Multi-Voice Recognition
  - Support Promotion and En Passant Captures

TIMELINE & MILESTONES

Feature	Owner	Date
Valid Sample Detection	Varun	19-Nov
Complete Keyboard Encoder	Michael	24-Nov
Complete Keyboard Entry -> Screen Capability	Michael	24-Nov
Time Warping Demonstrated	Varun	24-Nov
Complete Move Checker	Michael	26-Nov
Complete Chess Engine and Board Representations	Michael	3-Dec
Complete Chess Graphics Capability	Michael	3-Dec
Memories Instantiated	Varun	3-Dec
One DTW Engine Fully Functional	Varun	3-Dec
Whole DTW System Functional	Varun	5-Dec
Integrate System	M & V	7-Dec

Questions?