



By Kyle Alspach, Grant Parker, and Robert Brosig

How it works

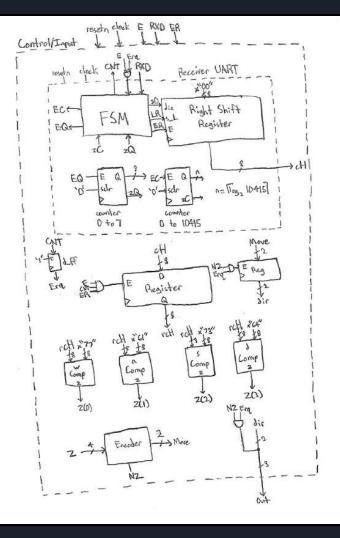
- Receiver to accept input from a computer, serially at 9600 baud, across a USB
- PuTTY is used to transfer this data
- Accepts only the characters w, a, s, and d, and only lowercase
- Output of receiver is a 3 bit encoded direction, with MSB being a pulse when a correct character is output, and the last two being the direction
- This is sent to a control circuit, and read by an FSM
- FSM generates 2 or 4 randomly and puts it on the grid
- FSM adds matching values when shifting in direction dictated by receiver, and also checks to see if any of the values equal 2048 (victory condition)
- It is possible to lose the game if all the tiles are full and cannot be added together
- Grid is constantly outputted to a file that makes it compatible for a VGA, then sent to a VGA



The Game

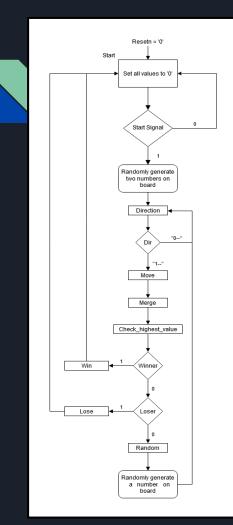


- 2048 is a game where the user slides tiles in a direction, and matching tiles add up.
- Random numbers spawn with each slide
- If the table fills, they lose, if they reach 2048, they win
- The game was created in 2014 by Gabrielle Cirulli as a web game
- Was based on "Threes", which was a much slower paced game released the same year



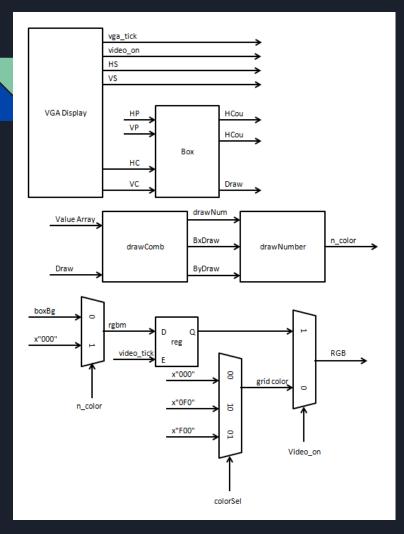
Receiver Datapath

- Composed of UART, comparators, an encoder, registers, and logic gates
- Output earlier described can be seen
- The characters w, a, s, and d that are compared to are stored similarly to an LUT



FSM

- Main controller of the game
- Determines number movement and merging
- Stores values into an array



VGA Display

- Uses a VGA connection to output to a monitor
- Objects are drawn on the screen depending on the Horizontal and Vertical count.
- Kept in sync by HS and VS to keep images from moving.
- Screen is thought of as a graph and is told to color a certain area when a signal is high.



FSM Simulation

(0,16,32,1	(4,0,0,0),(0,0 (4,0,2,0),((0,0,0,0),(0,0,0,0),0		(4,0,2,0),(0,0,0,0)		(4,2,0,0),(0,0,0,0),(0,0,0,0),□		0,0,0),0	(4,2,0,0),(0,0,0,4)		(0,0,4,20	(0,0,4,2),	(0,0,0,4)	(4,0,4,2)		
2,4,16,2	0,2,0,0			0,0,0,0		2,0	2,0,4,0		0,0,2,4		2,0,2,4		2,2,4,0		8,0,0,0		
16,128,2,4	0,0,0,0																
4,16,64,8	0,0,0,0							4,0,0,0									
16,32,16,0	0,1	0,0,4	X	0,2,0,4				0,0,2,4			χ.		χ	2,4,0,0		2,4,0,4	
lose	stD	direction	move	merge	check_hi0	random	direction	move	merge	check_hi0	random	direction	move	merge	check hill	xandom X	
5	5						4					6 7					
(0,16,32,1	(0,0) (4,0,4,2), (0,0,0,4)0			X (0,0,0,0), (0,0,0,2), (0,0,0,4), D X (0,4,0			(0,4,0,2)	, (0,0,0,2) (4,4,4,2 (4,4,4,16)			,(0,0,0,00 (4,4,4,16),(, (0,0,0,00) (4,4,4,10) (8,4		(8,4,16,0)	,(0,0,0,00	
2,4,16,2	8,0,0,0			8,4,0,4					8,0,0,0	0,0,0,0 2,		0,4,0		0,0,2,4	0,0,2,4		
16,128,2,4		0,0,0,0		(4,0,0,0)					0,0,0,0								
4,16,64,8		4,0,0,0		2,0,0,0						0,0,0							
16,32,16,0	2,40 2,4,0,4			χ 0,0,0,0 χ			2,	0,4,0 (2,4,4,4)		X	16,4,4,4					0,16,4,8	
lose	cheD	random	direction	move	merge	check_hi	random	direction	move	merge	check hill	random	direction	move	merge	check hill	
5	6 7					Y	5				Y I	4					
	2,4,16,2 16,128,2,4 4,16,64,8 16,32,16,0 lose 5 (0,16,32,11 2,4,16,2 16,128,2,4 4,16,64,8 16,32,16,0 lose	2,4,16,2 16,128,2,4 4,16,64,8 16,32,16,0 Iose 5 (0,16,32,11 2,4,16,2 16,128,2,4 4,16,64,8 16,32,16,0 16,128,2,4 4,16,64,8 16,32,16,0 2,40 cheD (cheD	2,4,16,2 16,128,2,4 4,16,64,8 16,32,16,0 10se 5 (0,16,32,11 2,4,16,2 16,128,2,4 (0,	2,4,16,2 16,128,2,4 4,16,64,8 16,32,16,0 10se 5 (0,16,32,11 (0,□) (4,0,4,2), (0,0,0,4)□ 2,4,16,2 16,128,2,4 (0,□) (4,0,4,2), (0,0,0,4)□ 2,4,16,2 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0,4 16,128,2,4 0,0,0,0,4 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 0,0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 0,0,0 16,128,2,4 16,22,10 0,0,0 16,128,2,4 16,22,10 0,0,0 16,128,2,4 16,22,10 0,0,0,0 16,128,2,4 16,22,10 16,22,1	2,4,16,2 16,128,2,4 4,16,64,8 16,32,16,0 0,0,0,0,4 10se 5 (0,0,0,0,4 10se 5 (0,0,0,0,4 10se 5 (0,0,0,0,4 10se 5 (0,0,0,0,0) 4,16,64,8 (0,0,0,0,4) (0,0,0,0,0) (0,0,0,0,0) (0,0,0) (0	2,4,16,2 0,2,0,0 0,0,0,0 16,128,2,4 4,16,64,8 16,32,16,0 0,0,0,4 0,2,0,4 10se st□ direction move merge check hi□ 5 (0,16,32,11 (0,□ (4,0,4,2),(0,0,0,4)□ ((0,0,0,0),(0,0,0,2),(0,0,0)) (0,0,0,0,0) (0,0,0,0) (0,0,0,0), (0,0,0,0), (0,0,0,0), (0,0,0,0) 16,128,2,4 0,0,0 (0,0,0,0) 16,128,2 (0,	2,4,16,2 0,2,0,0 0,0,0,0 2,0 16,128,2,4 0,0,0,0,0 2,0 16,32,16,0 0,0,0,0,4 0,2,0,4 10se st□ direction move merge check hi□ random 5 5 (0,0,0,0,0) 2,0 16,128,2,4 0,0,0,0,0 1cse st□ direction move merge check hi□ random 5 5 (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	2,4,16,2 0,2,0,0 0,0,0,0 2,0,4,0 16,128,2,4 0,0,0,0 2,0,4,0 16,32,16,0 0,0,0,4 0,2,0,4 0se st□ direction mexge check hi□ random (0,16,32,11 (0,0,0,4) 0,0,0,0) (0,4,0,2) (0,0,0,0) 16,128,24 0,0,0,0 0,0,0,0 (0,4,0,2) 16,32,16,0 0,0,0,0,0 0,0,0,0) (0,4,0,2) (0,16,32,11) (0,0,0,0,0) (0,0,0,0) (0,4,0,2) (16,128,2,4) 0,0,0,0 8,4,0,4 (0,0,0,0) (16,128,2,4) 0,0,0,0 2,0,0,0 4,0,0,0 (16,128,2,4) 0,0,0,0 2,0,0,0 2,0,0,0 (16,128,2,4) 0,0,0,0 2,0,0,0 2,0,0,0 (16,128,2,4) 0,0,0,0 2,0,0,0 2,0,0,0 (16,128,2,4) 0,0,0,0 2,0,0,0 2,0,0,0 (16,64,8) 4,0,0,0 2,0,0,0 2,0,0,0 (16,64,8) 2,40,0,4 0,0,0,0 2,2,0,0,0 (16,82,16,0) 2,40,0,4 0,0,0,0,0 2,2,0,0,0 (16,82,16,0)	2,4,16,2 0,2,0,0 0,0,0,0 2,0,4,0 16,128,2,4 0,0,0,0 2,0,4,0 4,16,64,8 0,0,0,0 0,0,0,0 16,32,16,0 0,0,0,4 0,2,0,4 0se st□ direction move merge check hi□ random direction move 5 5 (0,0,0,0,0) 8,4,0,4 16,128,24 0,0,0,0,0 2,4,16,2 8,0,0,0 16,128,24 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	2.4,16,2 0,2,0,0 0,0,0,0 2,0,4,0 0,0,2,4 16,128,2,4 0,0,0,0 0,0,0,0 0,0,0,0 4,16,64,8 0,0,0,0 0,0,0,0 16,32,16,0 0,0,0,4 0,2,0,4 0,16,32,16 0,0,0,0,0 0,0,0,0 10se stD direction move 5 5 4 (0,16,32,11 (0,0,0,0,0) (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	24,16,2 0,2,0,0 0,0,0,0 2,0,4,0 0,0,0,2,4 16,128,2,4 0,0,0,0 0,0,0,0 2,0,4,0 0,0,0,0 4,16,64,8 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 16,32,16,0 0,0,0,4 0,2,0,4 0,0,0,2,4 0,0,2,4 lose stD direction move merge check hiD xandom direction move merge check hiD 5 5 4 (0,0,0,0,4) (0,0,0,0), (0,0,0,2), (0,0,0,0,0), (0,0,0,2), (0,0,0,0,0), (0,0,0,2), (0,0,0,0,2), (0,0,0,0,2) (4,4,4,2D) (4,4,4,4)	24,16,2 0,2,0,0 0,0,0,0 2,0,4,0 0,0,2,4 2,1 16,128,2,4 0,0,0,0 2,0,4,0 0,0,0,0 2,0,4,0 2,0,0,0 4,16,64,8 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 16,32,16,0 0,0,0,4 0,2,0,4 0,0,2,4 0,0,2,4 0se stD direction move merge check hiD random 5 5 4 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 2,4,16,2 8,0,0,0 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	24,16,2 0,2,0,0 0,0,0,0 2,0,4,0 0,0,2,4 2,0,2,4 16,128,2,4 0,0,0,0 0,0,0,0 2,0,4,0 0,0,0,0 4,16,64,8 0,0,0,0 0,0,0,0 0,0,0,0 0,0,0,0 16,32,16,0 0,0,0,4 0,2,0,4 0,0,2,4 0,0,2,4 0se stD direction move merge check hiD random 5 5 4 0,0,0,0 (0,4,0,2), (0,0,0,4) (0,0,0,0,0) (4,4,4,16), (0,0,0,0) 2,4,16,2 8,0,0,0 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	24,162 0,2,0,0 0,0,0,0 2,0,4,0 0,0,2,4 2,0,2,4 2,2,2,4,0 16,128,24 0,0,0,0 0,0,0,0 2,0,4,0 0,0,0,0 4,1 16,32,16,0 0,0,0,0,4 0,2,2,0,4 0,0,0,2,4 4,1 16,32,16,0 0,0,0,0,4 0,2,2,0,4 0,0,2,4 4,1 16,32,16,0 0,0,0,0,4 0,2,2,0,4 0,0,2,4 4,1 16,32,16,0 0,0,0,0,4 0,2,2,0,4 0,0,2,2,4 4,1 (0,16,32,17 (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	24,162 0,2,0,0 0,0,0,0 2,0,4,0 0,0,2,4 2,0,2,4 2,2,2,4,0 16,128,24 0,0,0,0 0,0,0,0 4,0,0,0 4,0,0,0 41,16,648 0,0,0,0 0,0,2,4 2,0,2,4 2,2,2,4,0 16,32,160 0,0,0,0,4 0,2,2,0,4 0,0,2,2,4 2,4,0,0 10se stD direction move merge check hiD random 5 5 4 6 6 6 6 (0,10,4,2), (0,0,0,4)D (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	24,162 0,2,0,0 0,0,0,0 2,0,4,0 0,0,2,4 2,0,2,4 2,2,4,0 8,0,0,0 16,128,24 0,0,0,0 0,0,0,0 0,0,0,0 4,0,0,0 4,0,0,0 16,32,160 0,0,0,4 0,2,0,4 0,2,0,4 0,0,2,4 2,4,0,0 16,32,160 0,0,0,4 0,2,0,4 0,2,0,4 0,0,2,4 2,4,0,0 16,32,160 0,0,0,4 0,2,0,4 0,2,0,4 0,0,2,4 2,4,0,0 16,32,160 0,0,0,0,4 0,2,0,4 0,0,2,4 2,4,0,0 16,32,160 0,0,0,0,4 0,2,0,4 0,0,2,4 2,4,0,0 16,32,160 stD direction move merge check hiD xandom 2,4,162 8,0,0,0 (4,4,4,2D) (4,4,4,16) (0,0,0,0D) (4,4,4,10) (8,4,16,4) 16,128,24 0,0,0,0 4,0,0,0 8,4,0,4 8,0,0,0 0,0,0,0 2,0,4,0 0,0,0,2,2,4 16,128,24 0,0,0,0 4,0,0,0 2,0,4,0 0,0,0,0 0,0,0,0 2,0,4,0 0,0,0,0,0 16,32,160 2,4,0,4 0,0,0,0 2,0,4,0 0,0,0,0 <	

Btn_direction: 4 is right, 5 is down, 6 is left, 7 is right

Receiver Simulation

Name	Value	0.000000 us	500.000000 us	1,000.000000 us	1,500.000000 us	2,000.000000 us	2,500.000000 us	3,000.000000 us	3,500.000000 us	4,000.000000 us	4,500.000000 us	
1å resetn	1											
16 clock	0											
16 E	1											
🔓 Ereggy	1											
le nxd	1											
✓ ♥ sig[2:0]	0		0			0			X•X	0		
16 [2]	0											
16 [1]	0											
16 [0]	0											
U clock_period	10000 ps	10000 ps										
🕌 chara	w						u .					
14 charc	J						3					
Name	Value	0.000000 us	500.000000 us	1,000.000000 us	1,500.000000 us	2,000.000000 us	2,500.000000 us	3,000.000000 us	3,500.000000 us	4,000.000000 us	4,500.000000 us	
🔓 resetn	1											
🔓 clock	0											
16 E	1											
🔓 Ereggy	1											
le rxd	1											
	0		0	XX		0			X	0		
	10000 ps	10000 95										
	w						8					
14 charc	j						i					

- Sig is the output, w and j are input by the simulation
- Emits a pulse with each input matching the internal LUT
- Ignores inputs that don't match the internal LUT
- Problem: Matching wrong clock pulse
- Resolved by moving S5 'done' output of UART to S1



References

• <u>https://2048game.info/the-history-of-2048/</u>

• <u>https://www.ibtimes.com/what-2048-convoluted-origin-threes-1024-game-clone-topping-app-store-charts-1568533</u>