Is the propeller chip fast enough to hang off the Atari bus and pretend to be RAN.RIN?

From the datasheet for 1MHz/2MHz clocks:

TACC(650/310) is the time between address/rw stabilizing and read-data must appear

TDSU(100/50) is the time until data stabilizes

THR(10/10) is the time data must be stable after

For writing the data appears on the bus earlier but can be read same as TDSU.

The Atari2600 runs off the color crystal at 1.19MHz (1/3). Closer to the 1MHz clock numbers.

The propeller runs at 80MHz. Each instruction takes 4 clocks: 20MHz instructions.

650@1MHz is roughly 547 at 1.19MHz. In propeller time that is 10 instructions.

The propeller would need to do this at a minimum to be a ROM:

* Read INA (address input and chip select)
* Shift/mask the address input (chip select to carry)
* If select, mask on outputs
* If not select, mask off outputs
* READ offset from shared
* Add address to offset
* READ value
* Shift value
* Write to output pins
* Jump to top of loop

Can I split this functionality into several COGs? Let one COG control the outputenable. Note that we get reads very fast in pulling in program bytes. But when we are processing the program we may not get a memory access.

Next step

* wires off of an Atari cartridge to a breadboard.
* Use the DoubleGap EPROM and an inverter on the breadboard
* Swap in the propeller chip. There may be voltage conversion issues.

JStella has a lot of promise. I can modify it to give me the debug spew I want. I’ll need a toggle to turn it on and off and I’ll need to define what I want in the way of spew.

ABCcba … background

O ball

M missile 0

m missile 1

P player 0

p player 1

## Missile Command

Talk about [ref] Z26 and its debug spew.

The missiles are all M0 (Missile 0). They can be single, double, or triple resulting in 1, 2, or 3 missiles. Two sets alternate every other frame giving 1 to 6 missiles, but at most 2 separate lines. Smart missiles are M0 but with no tracer. That allows the smart variety to move around and avoid the shot.

The cursor is the ball.

The shot from the base is M1 (Missile 1).

The explosions are always P1 (Player 1). The player shot is always M1 (Missile 1). There can be 3 of these so they alternate on 3 screens. If a missile is not in the air on its frame then it is rooted over the base.

Since the shots and explosions alternate every 3 and the missiles every 2 then the collisions with each shot are checked in turn. But collision detection is handled by hand anyway. You can disable the individual object in Z26. In combat a disabled missle does not collide when off. In asteroids and miscom they still do.

The score digits are player 1 and 2 combined. Each is tripled and interlaced so the digits are pPpPpP. The 1st digit is loaded during the horiz scan and then the other digits are written in before the tripling.

The bases are also player 1 and 2 tripled but not interlaced.

The far left digit is loaded into GRP0 on line 44. GRP0 and GRP1 are positioned on line 44 also. The duplicate of X on line 45 doesn’t seem to be visible. The same value is loaded into P1 and then P0. The last set of “x” on the end of each line seems to set the first digit for the next line. If you select game on the no-play you can get XX…X.

The shot status (group and current) is drawn with playfield A (the group) and P1 (the current).

44|b.....K..k..A..........|.....x..\*............................................|

45|.....y.......x.........|..................y..x..Y..X...................x..\*..|

46|.....y.......x.........|..................y..x..Y..X...................x..\*..|

47|.....y.......x.........|..................y..x..Y..X...................x..\*..|

48|.....y.......x.........|..................y..x..Y..X...................x..\*..|

49|.....y.......x.........|..................y..x..Y..X...................x..\*..|

50|.....y.......x.........|..................y..x..Y..X...................x..\*..|

51|.....y.......x.........|..................y..x..Y..X.........\*...............|

(574 45 38 46) ( 56 64 19 79 50) nvbdIzC 7c 00 7c ff f800: a5 82 lda 82

(574 45 41 55) ( 56 64 19 79 50) nvbdIZC 00 00 7c ff f802: 85 1c sta GRP1

(574 45 44 64) ( 56 64 19 79 50) nvbdIZC 00 00 7c ff f804: 86 1b stx GRP0

(574 45 47 73) ( 56 64 19 79 50) nvbdIZC 00 00 7c ff f806: 84 1c sty GRP1

(574 45 50 82) ( 56 64 19 79 50) nvbdIZC 00 00 7c ff f808: 84 1b sty GRP0

90|O..................T...|...............K..\*..................................|

91|........z..............|...........y..........u.......\*......................|

92|O..................T...|.................K..\*................................|

93|........z..............|...........y..........u.......\*......................|

94|O..................T...|...............K..\*..................................|

95|........z..............|...........y..........u.......\*......................|

96|O..................T...|...............K..\*..................................|

97|........z..............|...........y..........u.......\*......................|

98|O..................T...|...............K..\*..................................|

99|........z..............|................y..........u.......\*.................|

100|O..................T...|.................K..\*................................|

101|........z..............|................Y..........u.......\*.................|

102|O............C.........|...T..................K..\*...........................|

103|........z..............|................Y..........u.......\*.................|

104|O...................t..|................K..\*.................................|

105|........z..............|................Y..........u.......\*.................|

224|O...................t..|................K..\*.................................|

225|........z..............|...........y..........u......\*.......................|

226|..............u....t..t|..F.....D.....C..c..c........t..u..0....N..n..\*......|

227|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

228|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

229|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

230|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

231|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

232|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

233|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

234|....2..............X...|......X..+..X..............Y.......Y....Y........\*...|

235|....2..............X...|......X..+..X..............Y.......Y....Y......\*.....|

236|..n.....d..d....2..0..x|...................t.....c..D.........\*..............|

237|.....Y......0..........|..............0.......\*..............................|

238|.....Y......0..........|..............0.......\*..............................|

239|.....Y......0..........|..............0.......\*..............................|

240|.....Y......0..........|..............0......\*...............................|

241|...d....~.....y........|.....................................................|

242|.......................|.....................................................|

John Saeger and contributors Z26 An Atari 2600 Emulator

<http://www.whimsey.com/z26/z26.html>

Miss comm.

* Upper area
  + P0 score
  + P1 score
* Game area
  + P1 (3 interlaced) player shot cloud
  + M0 (2 interlaced) missiles/smart-bombs
  + M1 (3 interlaced) player shot
  + BALL player cursor
* Bases
  + PLAYFIELD left and right border and base
  + P0 left bases
  + P1 right bases
* Indicator
  + PLAYFIELD ground, shot set indicator
  + P1 shots in current set indicator