Atari Controllers

The Atari controllers are read as part of the Stage 2 VBLANK process. The encoded data is partially decoded and processed as shown in the subsections that follow.

Joysticks

Up to four Joystick controllers can be attached to the computer console, each with a 9-position joystick plus a trigger button.

J1 STICK0 - STICK3 [0278,4] -- Joystick position sense

The 4 Joystick position sense variables contain a bit-encoded position sense as shown below:

```
 7 6 5 4 3 2 1 0
+-+-+-+-+-+-+-+-+
|0 0 0 0|R|L|D|U|
+-+-+-+-+-+-+-+-+
```

where: R = 0 indicates Joystick RIGHT sensor true.
L = 0 indicates Joystick LEFT sensor true.
D = 0 indicates Joystick DOWN sensor true.
U = 0 indicates Joystick UP sensor true.

Nine unique combinations are possible, indicating the possible Joystick positions shown below:

```
CENTER       $0F = 1111
UP           $0E = 1110
UP/RIGHT     $06 = 0110
RIGHT        $07 = 0111
DOWN/RIGHT   $05 = 0101
DOWN         $0D = 1101
DOWN/LEFT    $09 = 1001
LEFT         $0B = 1011
UP/LEFT      $0A = 1010
```

J2 STRIG0 - STRIG3 [0284,4] -- Joystick trigger sense

The four Joystick trigger sense variables each contain a single bit indicating the position of the Joystick trigger as shown below:

```
 7 6 5 4 3 2 1 0
+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0|T|
+-+-+-+-+-+-+-+-+
```

where: T = 0 indicates trigger pressed.

Paddles

Up to eight paddle controllers can be connected to the computer, each with a potentiometer and a trigger sense.

J3 PADDL0 - PADDL7 [0270,8] -- Paddle position sense

There is a single-byte variable associated with each paddle position sense; the values range from 228 for full counterclockwise rotation to 1 for full clockwise rotation.

The paddle values are often converted by the user, as shown below, to give a result of 0 for full counterclockwise rotation and 227 for full clockwise rotation:

```
VALUE := 228 - PADDLX;
```

J4 PTRIG0 - PTRIG7 [027C,8] -- Paddle trigger sense

The 8-paddle trigger sense variables each contain a single bit indicating the position of the paddle trigger as shown below:

```
 7 6 5 4 3 2 1 0
+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0|T|
+-+-+-+-+-+-+-+-+
```

where: T = 0 indicates trigger pressed.

Light Pen

The OS reads the position of a single light pen and stores the horizontal and vertical position codes in two variables; these codes are not the same as the actual screen coordinates. The pen position codes for different portions of the screen are shown below:

Left edge -- 67.
Codes increase in increments of one to a value of 227, then go to 0
and continue to increase monotonically (one count per color clock).
Right edge -- 7.

Upper edge -- 16.
Codes increase in increments of one (one count per two raster lines).
Lower edge -- 111.

The light pen hardware will read and latch the pen position 60 times per second, independent of the pen button position, which is separately sensed.

In order for the light pen to operate it must be positioned over a portion of the screen which has sufficient luminance to activate the photosensor in the pen; a blank (dark) screen will generally not provide enough luminance to utilize the light pen.

J5 LPENH [0234,1] -- Light pen horizontal position code

LPENH contains the horizontal position code for the light pen; the algorithm below (written in Pascal) shows the conversion from position code to screen coordinate (screen mode 7):

```
IF LPENH < 33              { check for rollover point }
     THEN                  { adjust values to right of rollover }
          XPOS := LPENH + 227
     ELSE                  { no adjustment to left of rollover point}
          XPOS : = LPENH;
XPOS := XPOS - 67;         {adjust for left edge offset}
IF XPOS < 0 THEN XPOS := 0;
IF XPOS > 159 THEN XPOS := 159;
```

J6 LPENV [0235,1] -- Light pen vertical position code

LPENV contains the vertical position code for the light pen; the algorithm below (written in Pascal) shows the conversion from position code to screen coordinate (screen mode 7):

```
YPOS := LPENV - 16;        {adjust for upper edge offset}
IF YPOS < 0 THEN YPOS := 0;
IF YPOS > 95 THEN YPOS := 95;
```

J7 STICK0 - STICK3 [0278,4] -- Light pen button sense

The light pen button sense is encoded in one of STICK0 - STICK3 (depending upon the actual controller port used) as shown below:

```
 7              0
+-+-+-+-+-+-+-+-+
|       |0|0|0|T|
+-+-+-+-+-+-+-+-+
```

where: T = 0 indicates the light pen button is pressed.

Driving Controllers

The driving controller has no position stops and thus allows unlimited rotation in either direction; the output of the controller is a 2-bit Gray code which can be used to determine the direction of rotation. The controller is sensed using the same internal hardware as the Joystick, thus the same data base variables are used for both.

J8 STICK0 - STICK3 [0278,4] -- Driving controller sense

The 4 driving controller sense variables contain an encoded rotation (position) sense value, as shown below:

```
 7 6 5 4 3 2 1 0
+-+-+-+-+-+-+-+-+
|0 0 0 0 1 1|val|
+-+-+-+-+-+-+-+-+
```

where a clockwise rotation of the controller produces the following continuous sequence of four values (shown in hexadecimal):

```
0F, 0D, 0C, 0E, 0F, 0D, ...
```

and a counterclockwise rotation of the controller produces the following continuous sequence of four values:

```
0F, 0E, 0C, 0D, 0F, 0E, ...
```

J9 STRIG0 - STRIG3 [0284,4] -- Driving trigger sense

The four driving trigger sense variables each contain a single bit indicating the position of the driving trigger as shown below:

```
 7 6 5 4 3 2 1 0
+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0|T|
+-+-+-+-+-+-+-+-+
```

where: `T = 0` indicates trigger pressed.