Pool Game Designed and Implemented Using Major-Minor FSM Setup

Group 10

Gary M. Matthias

Timothy Mwangi

Anthony Quivers

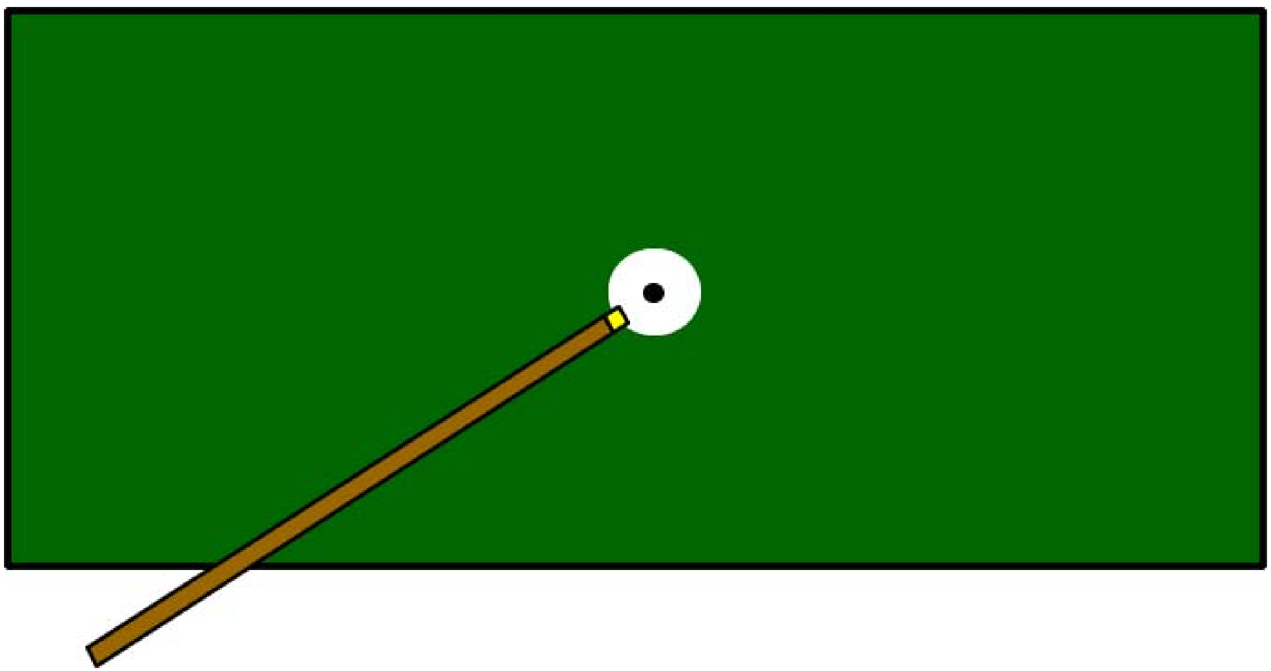
Input Module

- Physical Objects
  - Pool table
  - Pool stick
- Devices
  - Video camera
  - Accelerometer
  - Analog-to-digital converter

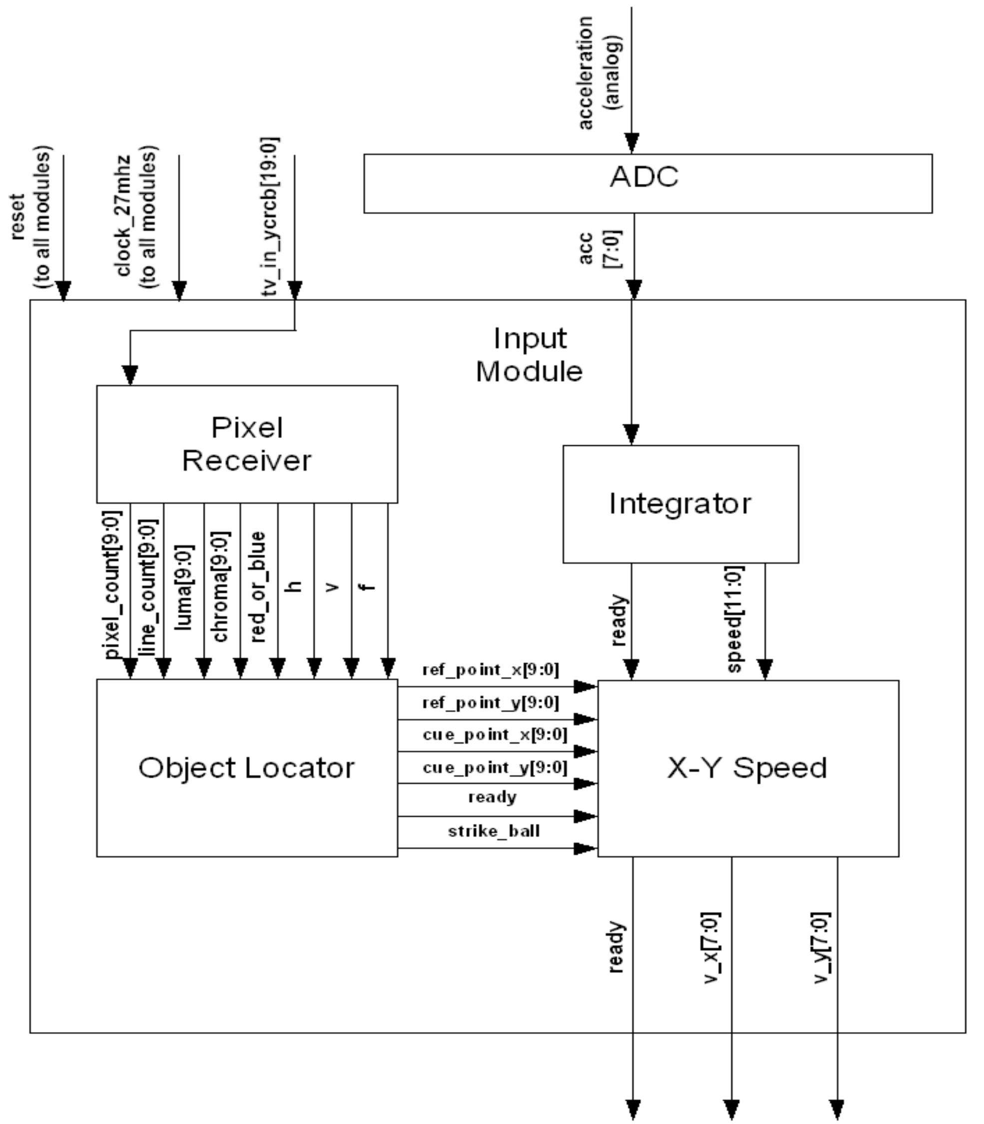
The Pool Stick

- Equipped with an accelerometer for speed calculations
- Colored at the tip for recognition by the camera

Camera View of Pool Table



Input Module: Block Diagram



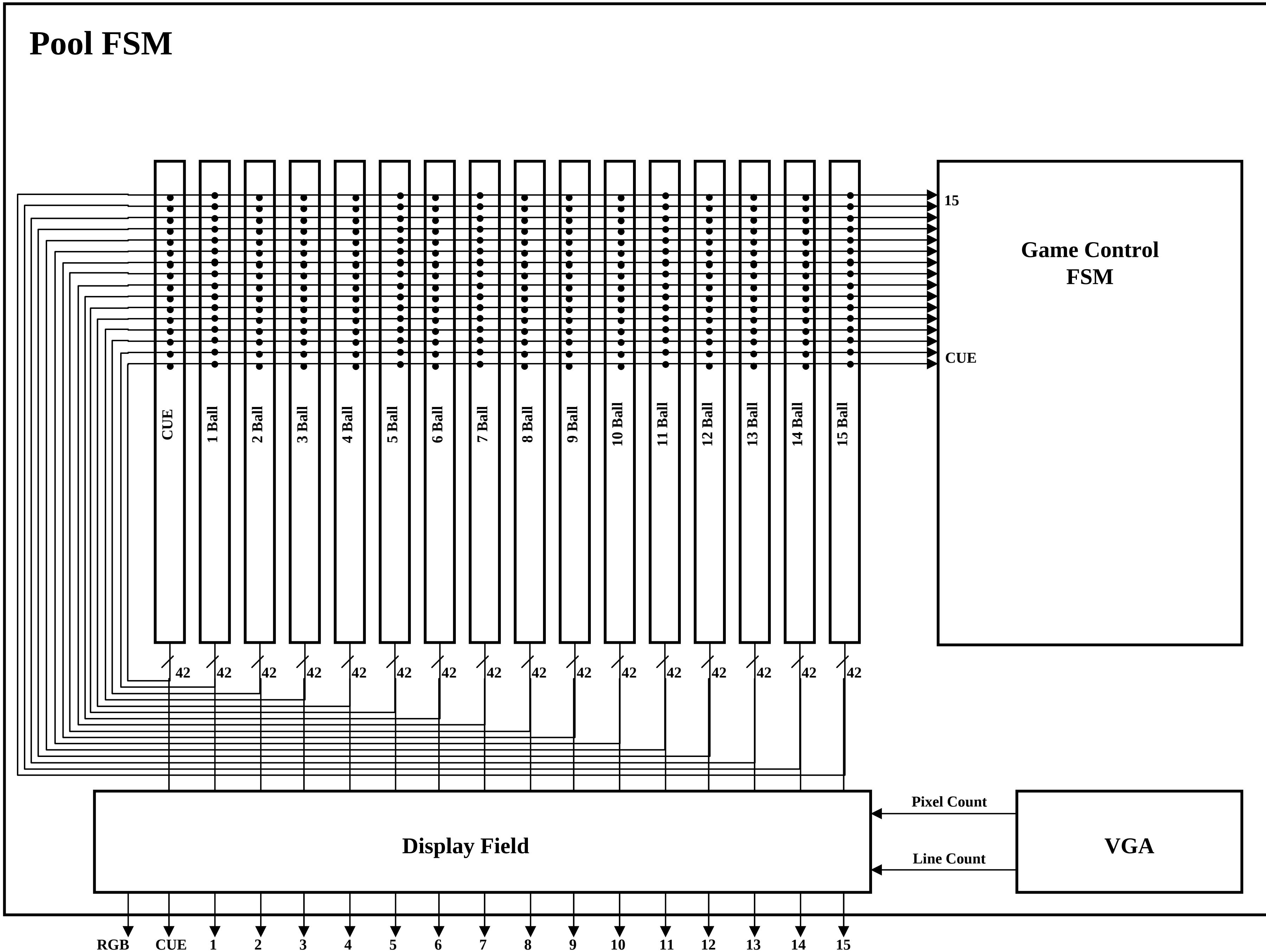
Game Logic Unit

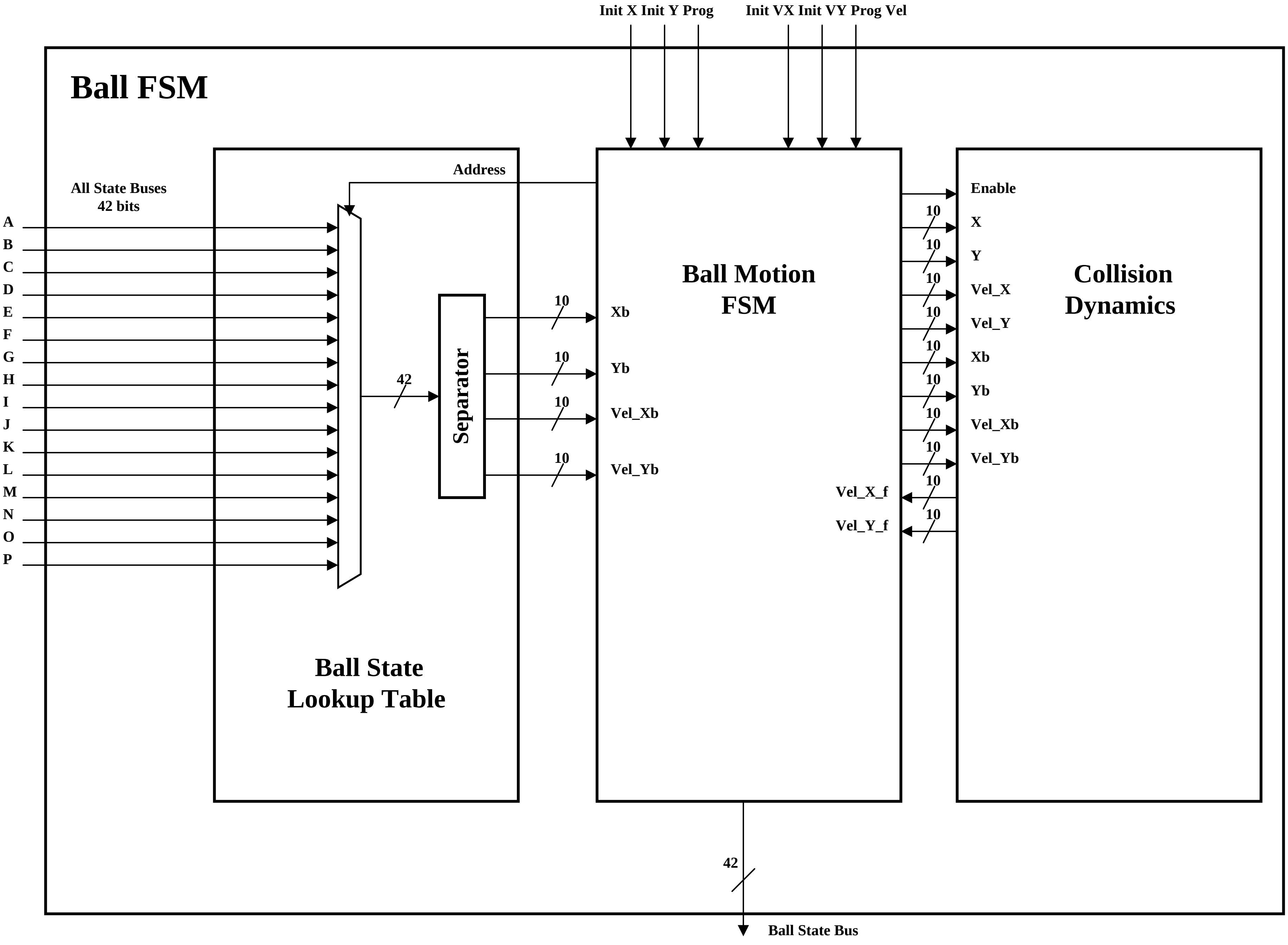
Design Specification

- Independent Set of modules used to control the all balls on the table.
- Responsible for controlling all events and enforcing all rules of 2 player British pool.
- Unit is abstracted away from user input interface and output graphics system.
- The positions of all balls are refreshed for the grapics module once per frame

Control Unit Implementation

- All balls on the table are controlled concurrently
- There are a total of 16 instances of **Ball FSM Modules**
  (i.e.: 1 for cue and 15 balls 1 through 15).
- Internal 2D VGA graphics interface for testing and debugging
  (This doubles as a backup graphics interface)


Pool FSM
15
CUE
Game Control
FSM
CUE
1 Ball
2 Ball
3 Ball
4 Ball
5 Ball
6 Ball
7 Ball
8 Ball
9 Ball
10 Ball
11 Ball
12 Ball
13 Ball
14 Ball
15 Ball
42
Display Field
Pixel Count
Line Count
VGA
RGB
CUE
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

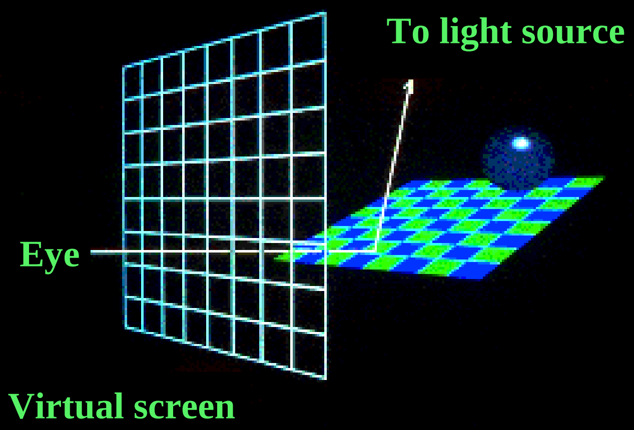


Init X Init Y Prog
Init VX Init VY Prog Vel
Ball FSM
All State Buses
42 bits
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Address
42
Separator
10
Xb
Yb
Vel_Xb
Vel_Yb
Ball State
Lookup Table
Ball Motion
FSM
Vel_X_f
Vel_Y_f
Enable
X
Y
Vel_X
Vel_Y
Collision
Dynamics
42
Ball State Bus

3-D Pool Display

- Perspective ray tracing
- The math
- Block diagram
- Memory requirements

Perspective Ray Tracing



The Math

Intersection of a line and plane

- Parametric equation of a line

$$x = x_1 + u\,(x_2 - x_1);\; y = y_1 + u\,(y_2 - y_1);\; z = z_1 + u\,(z_2 - z_1)$$

- Equation of a plane: $A\,x + B\,y + C\,z + D = 0$

Substitute equation of line into equation of plane

- A (x1 + u (x2 - x1))+B (y1 + u (y2 - y1))+C (z1 + u (z2 - z1))+D = 0

Solve for U

- U=Ax1+By1+Cz1+D/A(x1-x2)+/B(y1-y2)+C(z1-z2)

If A(x1-x2)+/B(y1-y2)+C(z1-z2) = 0

- Line is perpendicular to plane, no intersection/infinite intersections

Intersection of a line and sphere

- Equation of a sphere: $(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = r^2$

Define:

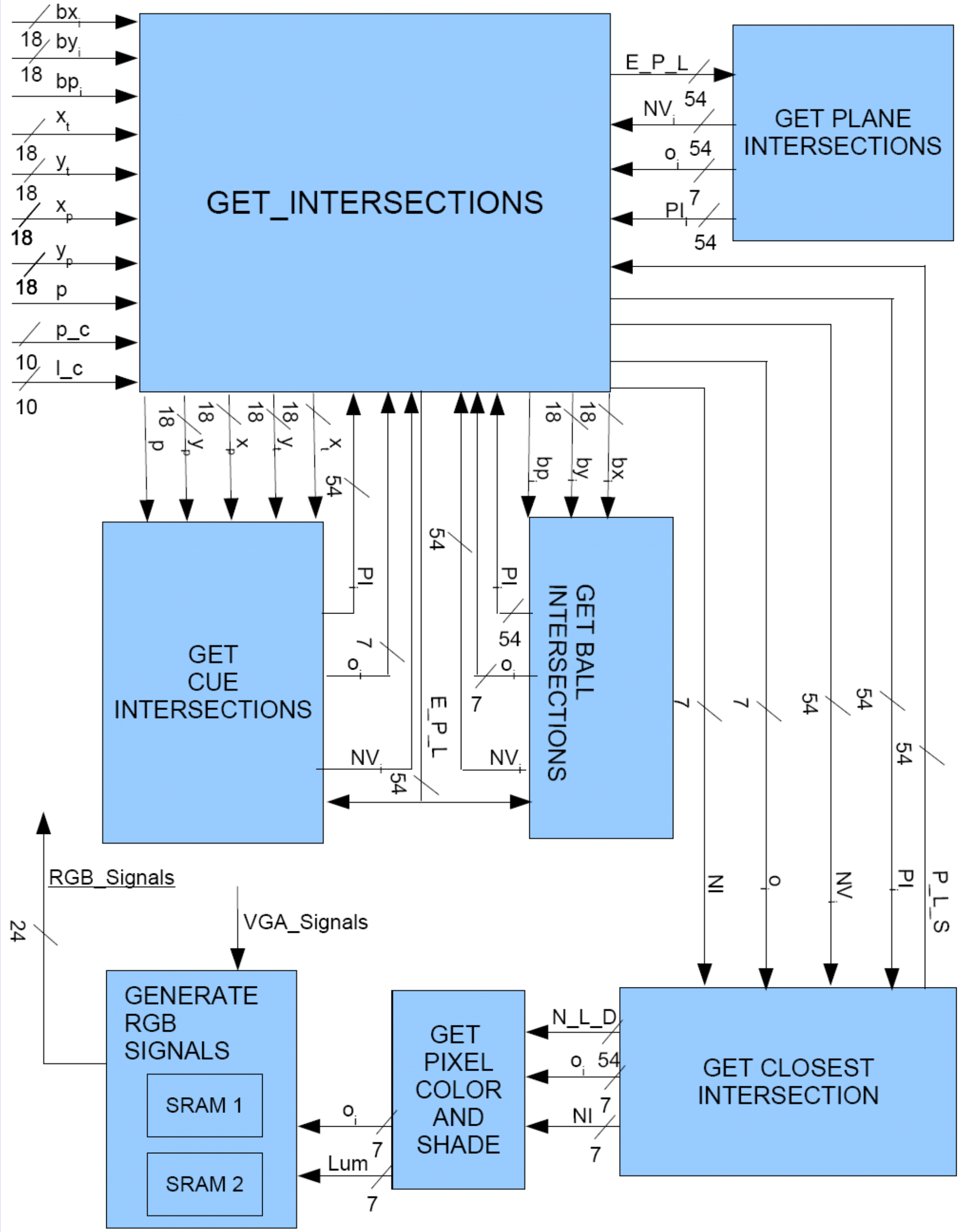
- $a = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$
- $b = 2[(x_2 - x_1) (x_1 - x_3) + (y_2 - y_1) (y_1 - y_3) + (z_2 - z_1) (z_1 - z_3)]$
- $c = x_3^2 + y_3^2 + z_3^2 + x_1^2 + y_1^2 + z_1^2 - 2[x_3 x_1 + y_3 y_1 + z_3 z_1] - r^2$

Quadratic equation: $au^2+bu+c = 0$; Solution: -b +-sqrt(b*b - 4ac)/2a

The exact behavior is determined by the expression within the square root :

b*b - 4 ac

- If this is less than 0 then the line does not intersect the sphere.
- If it equals 0 then the line is a tangent to the sphere intersecting it at one point, namely at $u = -b/2a$.
- If it is greater than 0 the line intersects the sphere at two points.



Graphics Module Block Diagram

Signal description

$Bx_{i,}$ By_i = x and y position of ball i

Bp_i = Boolean, Is ball i still on the table?

$X_{t,}$ Y_t = x and y position of cue tip

$X_{p,}$ Y_p = x and y position of pivot

P = Current player

p_c = pixel count

L_c = line count

E_P_L = Vector from eye to

pixel

Nv_i = Normal vector

O_i = Object

P_L_S = Vector from point to light source.

NI = Number of intersections

PI_i= Point of i^{th} intersection

N_L_D=dot product of normal at point and point to light vector.

Lum = Luminosity in YUV absolute color scale

Memory requirements

•For each pixel 0<=Luminosity<=120; Luminosity can be equal to 240 but in this 3-D renderer the maximum used is 120. –Requires 7 bits/pixel

•For each pixel $0 <= O_i <= 120$ is stored requires 7-bits/pixel

•Total memory requirement for double buffer:

$= ((7+7)/8) \times (1024 \times 768) \times 2 \times 10^{-6} = 2.76\text{MB}.$