Fast N-Body Simulation

Kade Phillips and Scott McCuen

N-body simulation

N-body problem—solving equations of motion for
N particles which interact with each other

Analytic solution generally intractable (or non-existent)
for N > 2

Why it’s hard

Direct simulation time complexity is $O(n^2)$

Normal solution—make approximations

Our solution—
dedicated, parallelized, application-specific hardware

Why use an FPGA?

Parallelization

Modularity

Goal: outperform a CPU

Modified Newtonian Dynamics

Gravitation	$F = GMm \div r^2$
Modified Inertia	$F = ma^2 \div a_0$
Acceleration	$a = (GMa_0)^{\frac{1}{2}} \div r$

Verlet Integration

$$x_{n+1} = 2x_n - x_{n-1} + a\Delta t^2$$

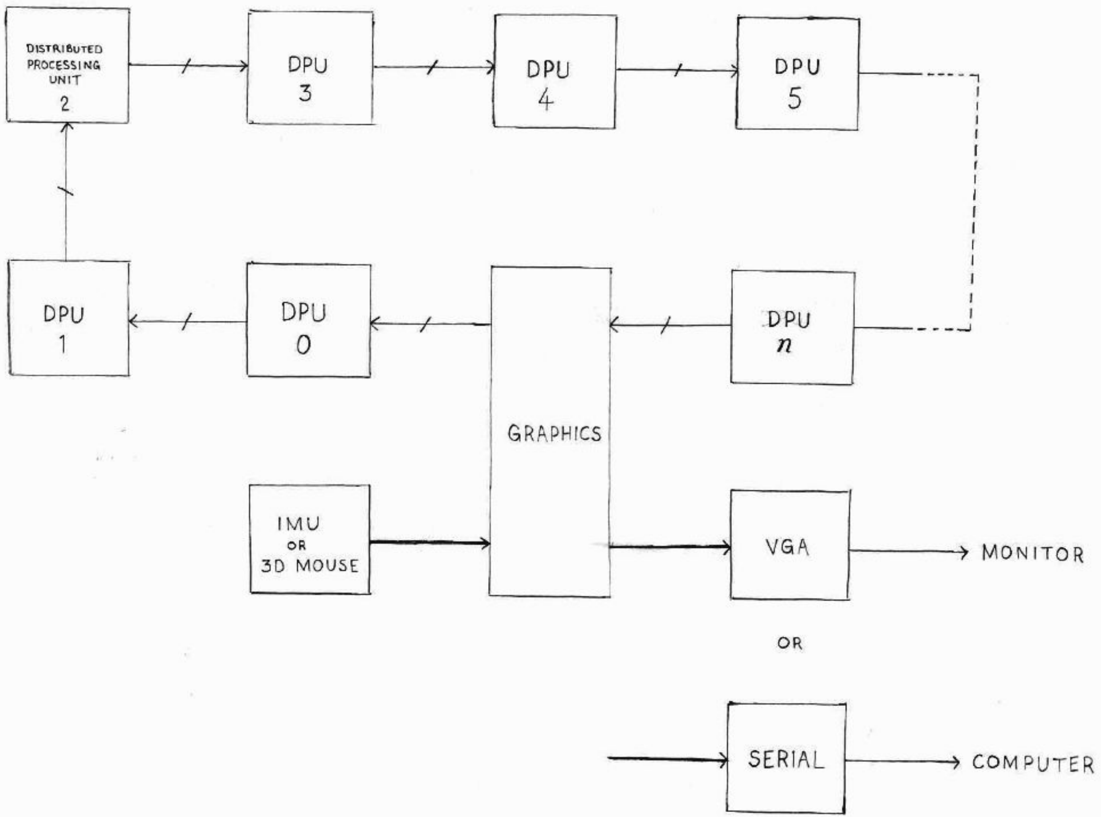
(It’s easy!)

C Implementation

```
int main(void) {
     FILE* disp = fopen("/dev/fb0", "w");
     uint32_t (*disp_buffer) = malloc(WIDTH*(MAX_Y+1) * sizeof(uint32_t));

     fputs("\e[2J", stdout);

     struct particle (*particles)[N] = malloc(2*N*sizeof(struct particle));

     ...
               long double dx = x1-x0;
               long double dy = y1-y0;
               long double dz = z1-z0;

               // unit length vector with direction of r: (vec1 - vec2) * 1/r
               //                 vector with length 1/r: (vec1 - vec2) * 1/r^2
               long double r_inv_squared = 1.0/(dx*dx + dy*dy + dz*dz);

               forces[jdx].x = dx*r_inv_squared;
               forces[jdx].y = dy*r_inv_squared;
               forces[jdx].z = dz*r_inv_squared;
```

video

PARALLEL BUS CARRIES ID/TAG AND THREE COÖRDINATE POSITION.



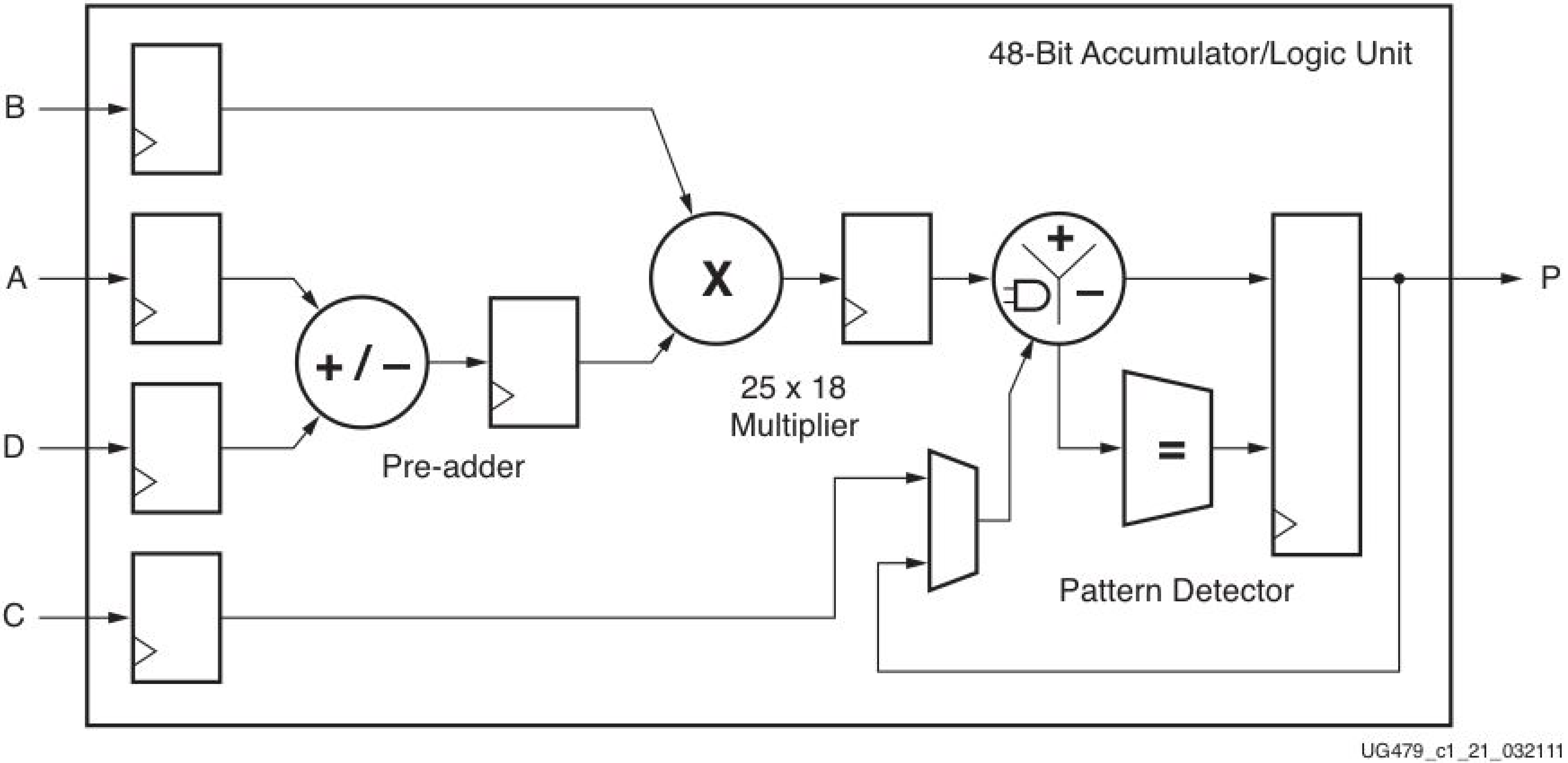
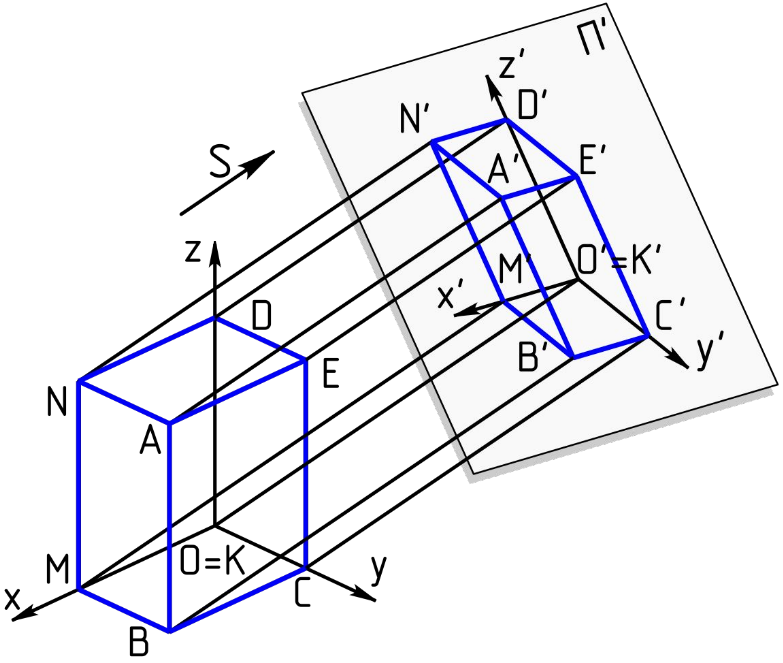
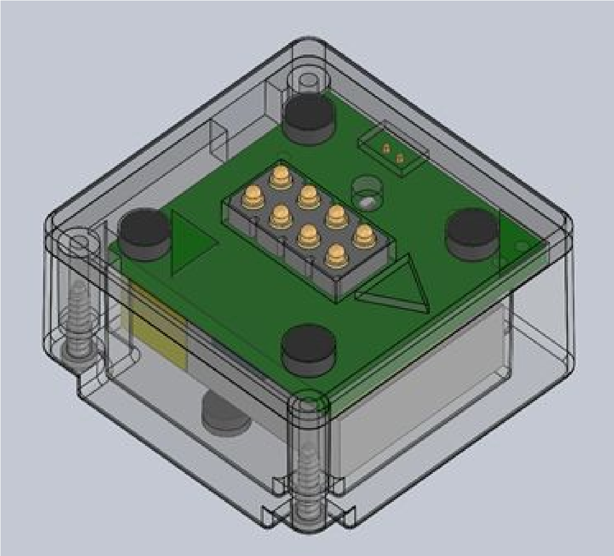


Figure 1-1: **Basic DSP48E1 Slice Functionality**

Display





Timeline

	5-11 Nov	12-18 Nov	19-25 Nov	26-2 Dec	3-9 Dec	10-13 Dec
Base DPU	K	K				
Optimized DPU		K	K			
Large N DPU			K	K		
2D display	S	S				
3D display		S	S			
3D interface			S	S		
Integration				K,S	K,S	
Demo, checkoff						K,S