Composing Design into Patterns

Henry’s Crypto Engine

Where is the memory? We think there is a specific location in system memory and the data is operated on a specific pinned page

Both a blocking and non-blocking call interface

Stateless Cipher and Hash Diagrams

But they usually have a running sum? They make you give the whole thing and give some state back??

Diagrams indicate physical connections not all possible logic connections

Network abstraction is fine for communication and arbitration so the diagram is simpler

How do we specify control? At one level it is control and another it is data

Top Level

What is the app doing?

Next Level

What does the machine look like?

Next Level

How do we implement those pieces?

One common pattern

Logically different things using the same common sub-blocks (time multiplexing patterns)

Share memory path and multiply unit

Ilia’s CPU (1999)

Fixed Function Pipeline

How to map textures onto polygons

Working on one frame at any point in time –parallelism across triangles
Fragments are pieces of pixels

UTL

Data from Host (vertex or texture data)

Host->network->MatMul->network->Cull->Raster->network->Shader->Fragment
Network->frame buffer

First level UTL doesn’t show parallelism—the parallelism should be in the second level

At the first level it is a stream processor and then you can add striping

Another pattern is the banked memory

There’s a general spectrum for programmable to fixed function-as you add more fixed functions the control ends up being complex and leads to programmable

Source of the data (Host) should be probably be a fifo to show its the input

A Single Shader

A single datapath for RGB and one for Alpha –both get data from input registers

A common pattern—multiple datapaths with one register file

Rima’s Audio Effects Processor/Amplifier

Fixed Function Pipeline

Serial Interface

Synchronous Dataflow

High Level UTL looks a lot like the block diagram

Micro Data Audio Processor Detail picture just adds some routing – it looks like it might be fixed function but it could be programmable

UTL Design

What are you trying to accomplish? Not just the hardware—but it is a highlevel abstract machine diagram so it does have some hardware decisions

However you could map different blocks from the UTL onto the same hardware block with time multiplexing

They are expressing machines not apps