

Microsoft Ignite をみた

@ogawa_tter さんに教えてもらった [Inside Azure Datacenter Architecture with Mark Russinovich](#) をみた。

1:04 あたりから FPGA 関連の話。

Cっぽいシンタックスの言語から, SystemVerilog 生成するらしい。

Quantum Inspired Algorithm の FPGA 実装のデモ。

紹介されてる言語はこんな感じか。

見えてる範囲だと {} の対応がとれてないけど, 一行目の行末は { なのかな。

たしかに, Cっぽい見た目処理記述しつつ, ハードウェア向けの属性も書ける, って感じだなあ。

```
[[max_threads(MAX_FIFO_DEPTH), async]] void ExecuteInstruction(uint 32 instrId_in, Instruction
instr_in, float32 beta_in, InstrFenceType sweepInstrFenceBase_in, b...
{
    uint32 randomVal = m_random.next();
    uint32 valve;

    if (!instr_in.m_instruction.m_accumulate)
    {
        float32 prob = from_uint<32>(randomVal);
        float32 recip = rcp(beta_in);
        float32 prob_tmp = sub<Denorm::Off>(log2(prob), 32.0);
        float32 valve_f = mul<Denorm::Off>(prob_tmp, recip);
        sim_assert(!t<Denorm::Off>(0.0, valve_f));
        auto convert = to_int<32>(valve_f);
        sim_assert(convert.is_valid);
    }
    if(!accumulate)
    {
        uint1[PipelinePerEngine] values;

        unrolled_for(const auto pipe_id : PipelinePerEngine)
        {
            EnergyType deltaEnergy = ALG::sum<EnergyType, EnergyType,
MAX_TERMS_IN_INSTRUCTION>(tempEnergy[pipe_id]);

            uint1 cur_val = cast<uint1>(independent_vals >> pipe_id);
            if(deltaEnergy <= 0)
            {
                // flipping must happen
                cur_val = cur_val;
            }
            else
            {
                if (deltaEnergy <= valve)
                {
                    cur_val = cur_val;
                    if (probing)
                    {
                        atomic
                        {
                            m_accept_incr_energy++;
                        }
                    }
                }
            }
            values[pipe_id] = cur_val;
        }
    }
}
```