



Challenges in Architecture Research

Joel Emer

Intel Fellow

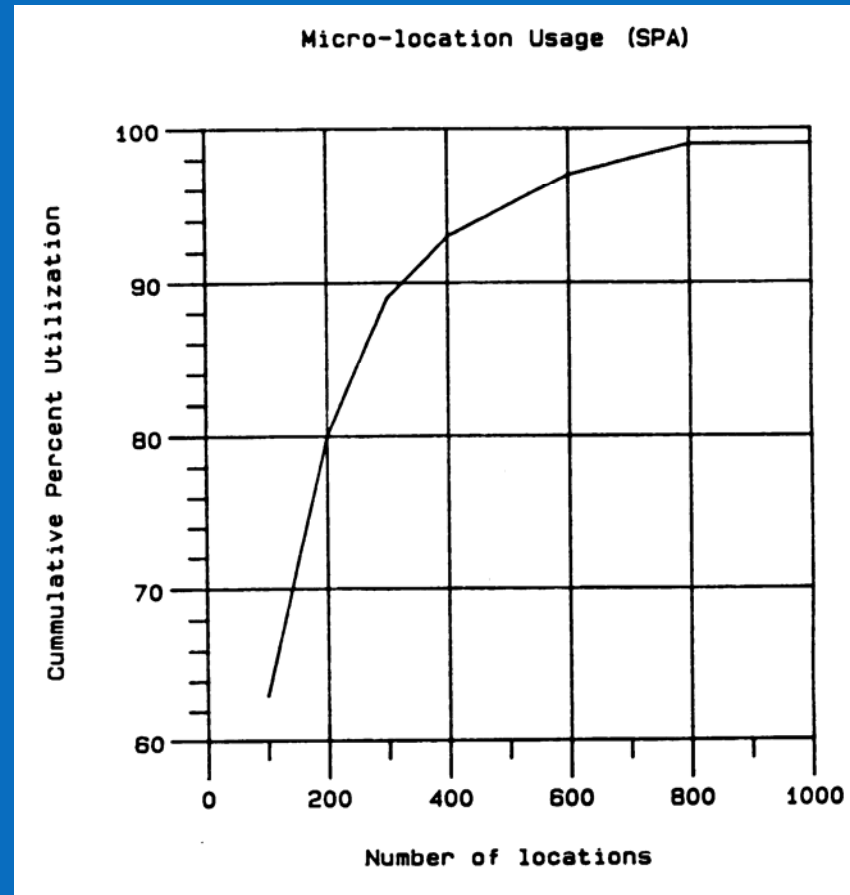
VSSAD

Iron Law of Performance

$$\text{Performance} = \frac{\text{Frequency} * \text{Instructions}}{\text{CPI}}$$

- Frequency – largely circuit design/technology
- CPI – largely organization
- Instructions – largely architecture/compiler

My Early Approach - 1981



uPC Histogram Chart – 1981-5

TABLE 8

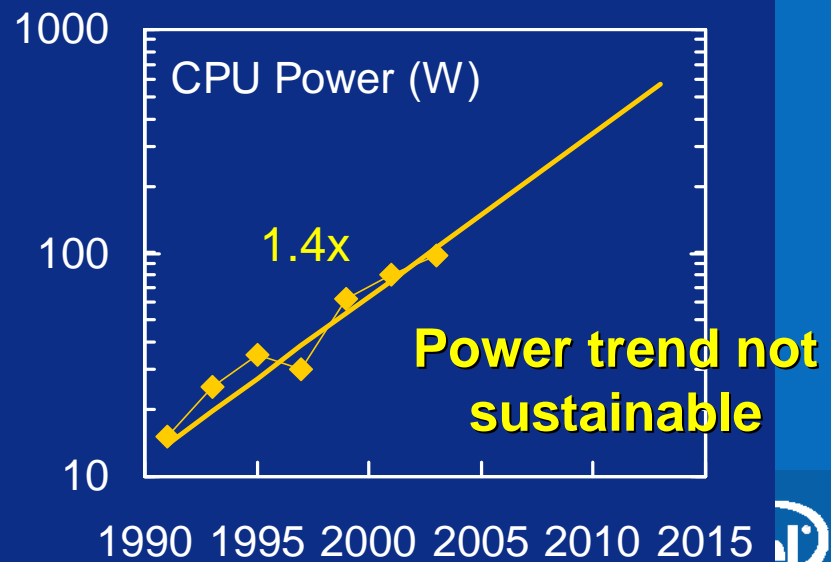
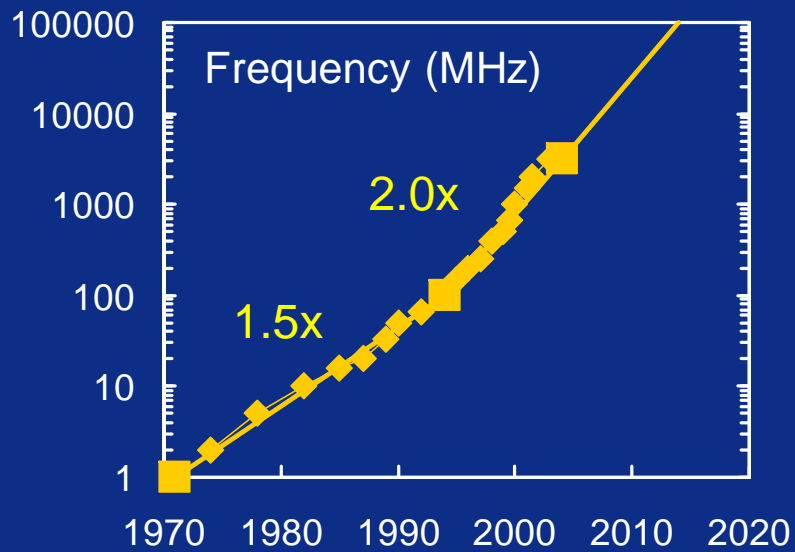
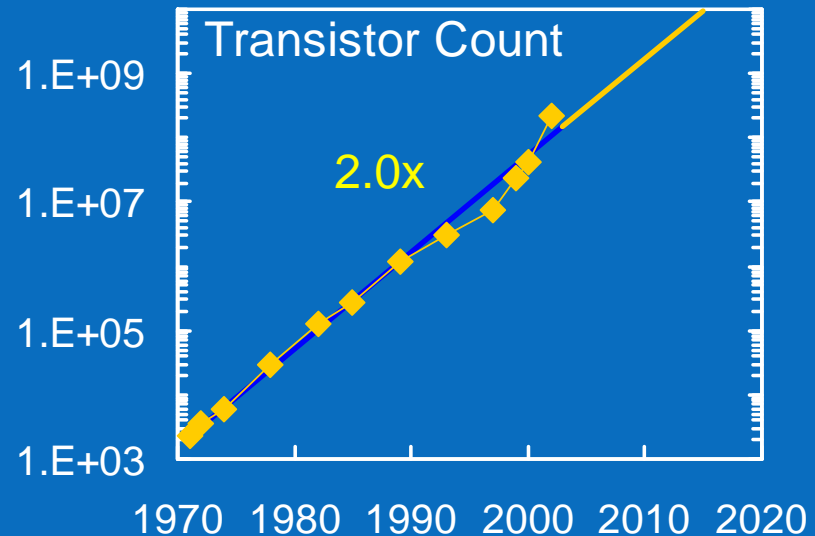
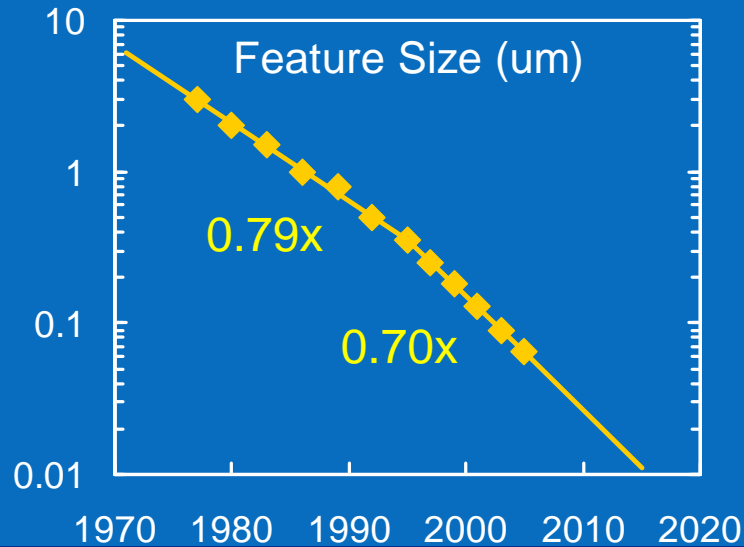
Average VAX Instruction Timing (Cycles per Instruction)							
	Compute	Read	R-Stall	Write	W-Stall	IB-Stall	Total
Decode	1.000					0.613	1.613
Spec1	0.895	0.306	0.364				1.565
Spec2-6	1.052	0.148	0.116	0.161	0.192	0.102	1.771
B-Disp	0.221					0.005	0.226
Simple	0.870	0.029	0.017	0.033	0.027		0.977
Field	0.482	0.049	0.058	0.007	0.002		0.600
Float	0.292	0.000	0.000	0.008	0.001		0.302
Call/Ret	0.937	0.133	0.074	0.130	0.184		1.458
System	0.434	0.015	0.031	0.014	0.028		0.522
Character	0.318	0.039	0.099	0.046	0.004		0.506
Decimal	0.026	0.002	0.000	0.001	0.002		0.031
Int/Except	0.055	0.002	0.005	0.004	0.006		0.071
Mem Mngm	0.555	0.061	0.200	0.004	0.003		0.824
Abort	0.127						0.127
TOTAL	7.267	0.783	0.964	0.409	0.450	0.720	10.593

Moore's Law - NOT

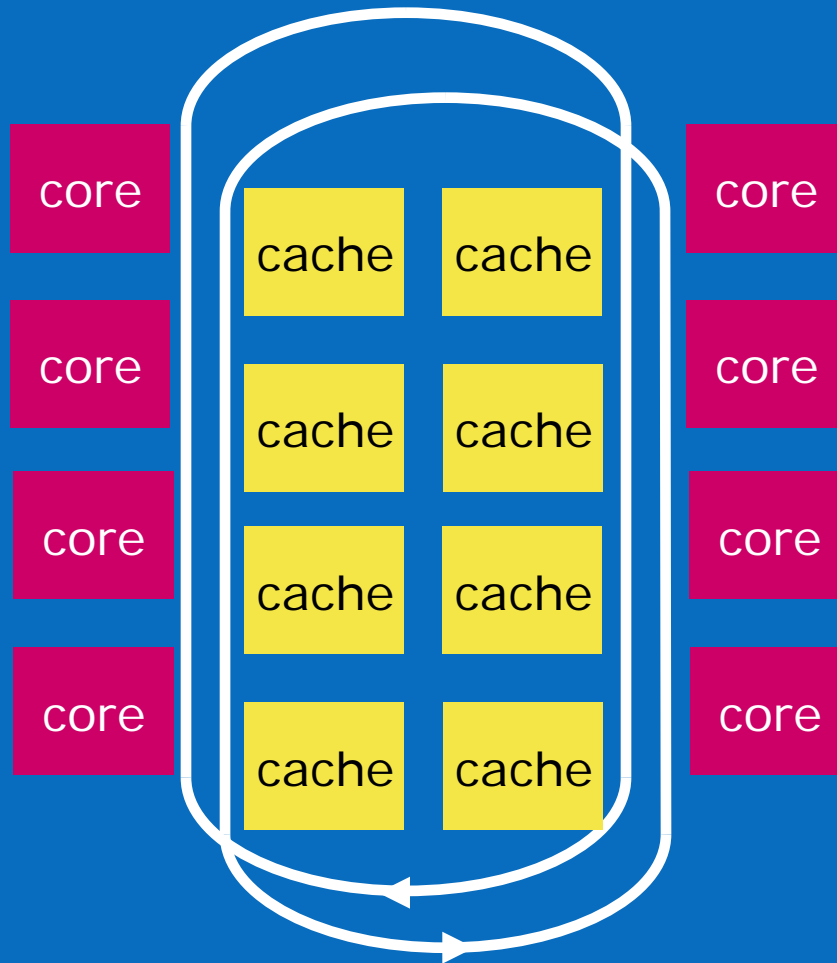


Photos from wikipedia

Moore's Law at Intel 1970-2005



Many-core processors



64 MB Cache

- 64 byte blocks
- 1M blocks

Replace each block 10X

- 4 miss / 1000 instructions
- 2.5B instructions

Simulation

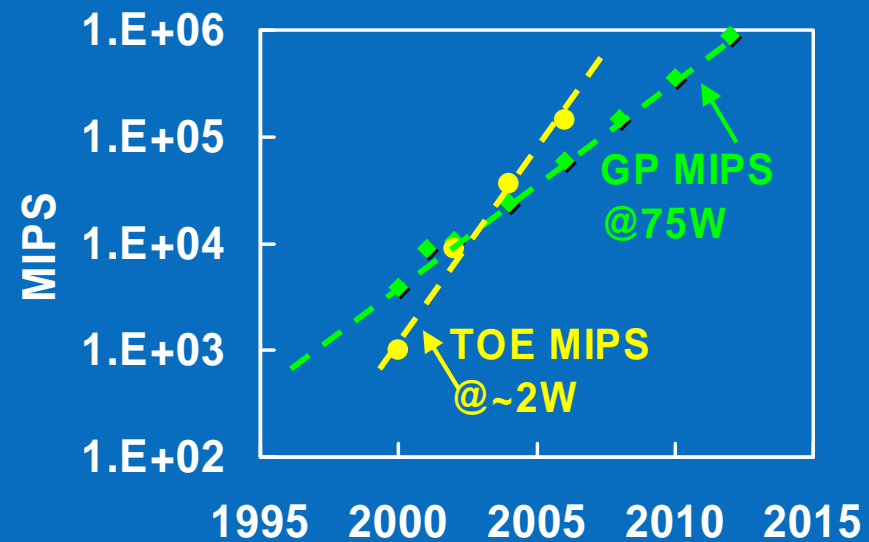
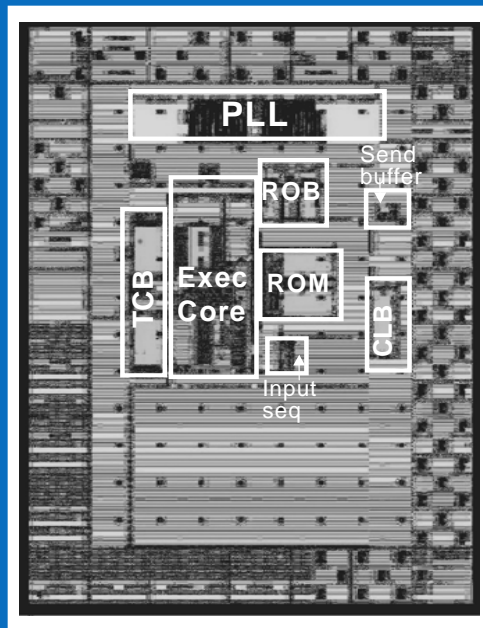
- 10K instructions/sec
- over 2.5 days

Software-based simulation is slow

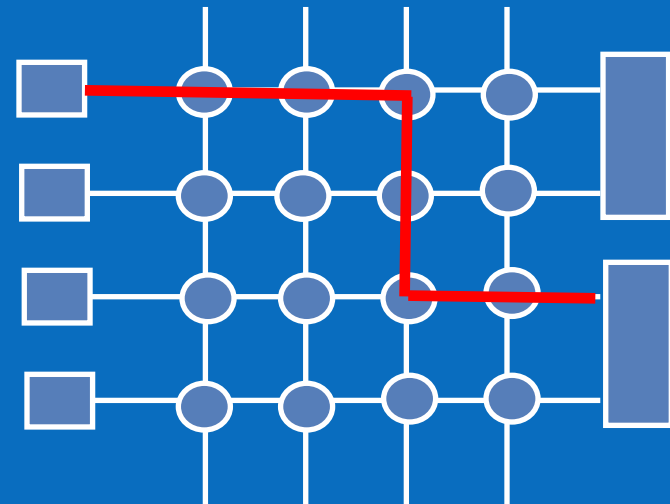
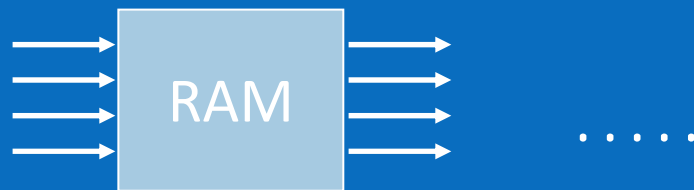
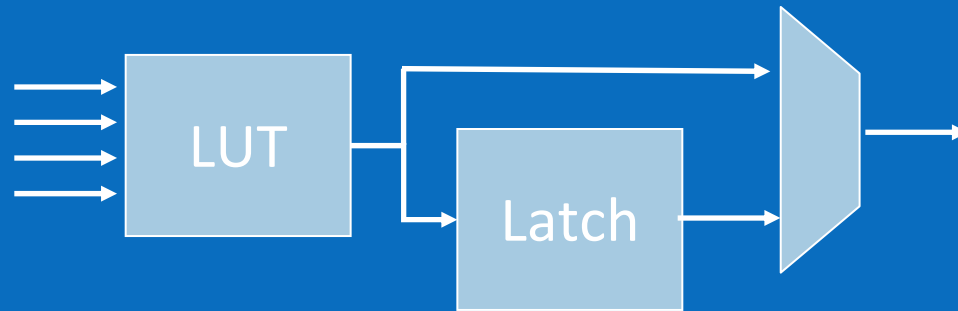
10Khz means ~100ms in 4 hr

- So new designs are often:
 - More incremental
 - Based on old (proven) ideas
 - Competitive reactions

Special purpose vs. General purpose gap



Field Programmable Gate Arrays



Hardware-based Simulation

FPGA CC	FET	DEC	EXE	MEM	WB
0	A	NOP	NOP	NOP	NOP
1	A				
2	A				
3	B	A	NOP	NOP	NOP
4	B	A			
5		A			
6	C	B	A	NOP	NOP
7		B			
8	D	C	B	A	NOP
9	D				
10	D				

= model cycle



FPGA Performance Model Metrics

FPGA cycle to Model cycle Ratio (FMR):

$$FMR = \frac{Cycles_{FPGA}}{Cycles_{Model}}$$

Simulator Frequency:

$$Frequency_{simulator} = \frac{Frequency_{FPGA}}{FMR}$$

FPGA Modeling

- The FPGA Modeling Promise
 - Raw speed of today's FPGAs: 100-400 MHz
 - Estimated FMR: 10
 - Frequency Simulator = $100 / 10 \rightarrow 10$ MHz
- The FPGA Modeling Trap
 - Hardware design is hard

Simulation Tradeoffs

Speed of Simulation



Accuracy

Modeling
Time

Simulation Tradeoffs

Speed of Simulation

Accuracy

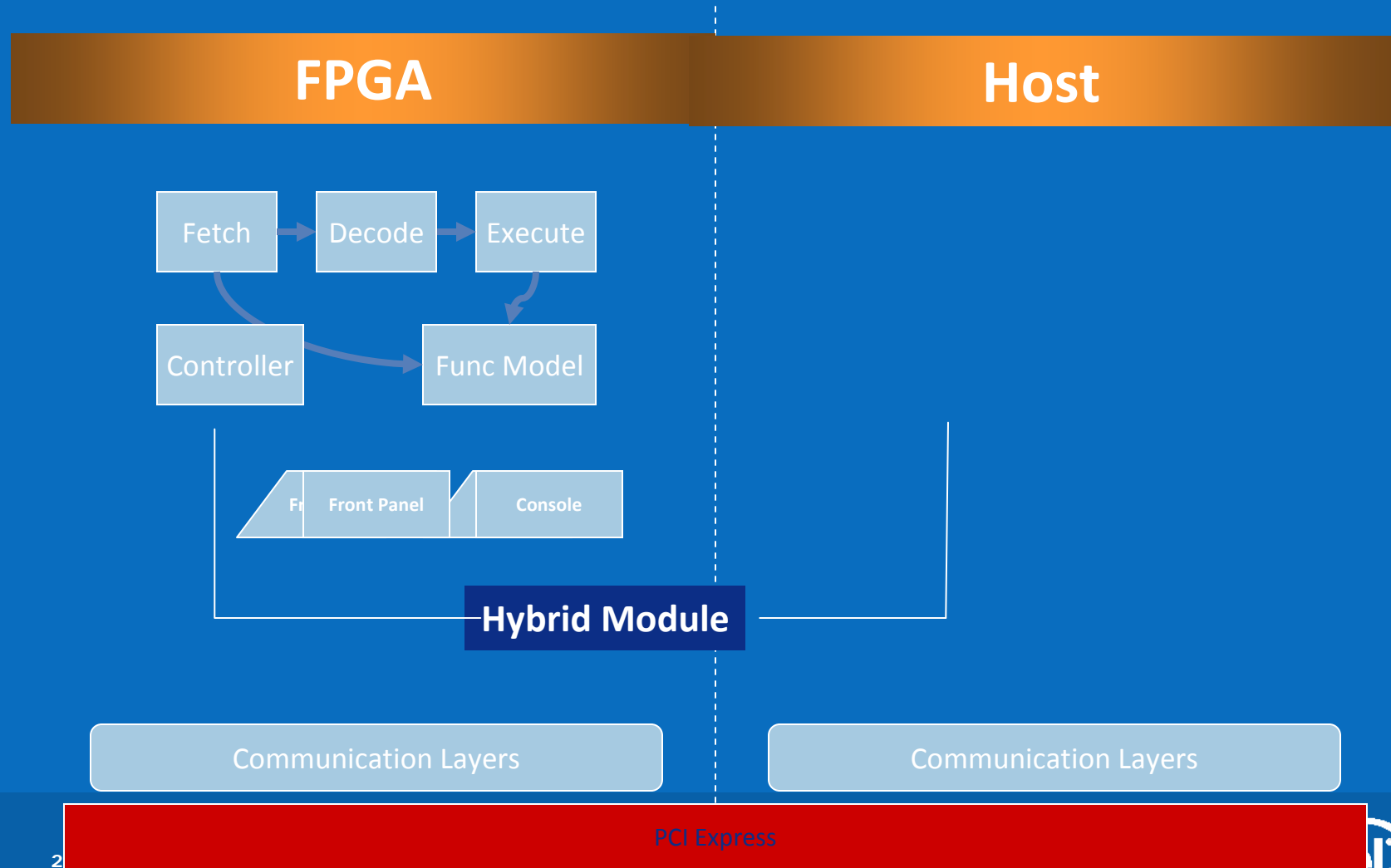


Reducing Development Time

- Employ modularity
- Separate micro-architectural and architectural behavior
- Separate behavior and timing
- High level synthesis (Bluespec)
- Use software/hardware hybrid...

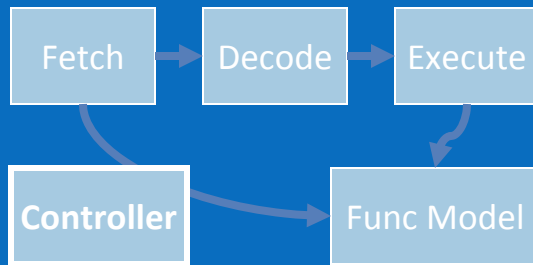


FPGA to Host: *Hybrid Modules*

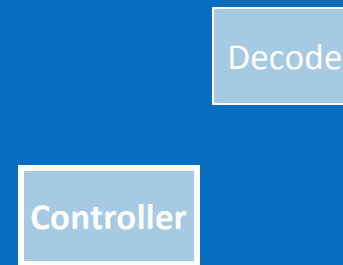


Hybrid Module Communication Paradigm

FPGA Modules



Software Modules



Server
- Accept Request
- Send Response

{ Start
Stop
Step



Client

{ PrintEvent
PrintStat } →

→ { PrintEvent
PrintStat }



RRR Specification Language

```
// -----  
// typedefs  
// -----  
typedef MSG_CLASS { HOST: UINT8,  FPGA: Bit#(8)  };  
typedef STATUS   { HOST: bool,   FPGA: Bool    };  
typedef UINT     { HOST: UINT64,  FPGA: Bit#(64) };  
  
// -----  
// create a new service called CONTROLLER  
// -----  
service CONTROLLER  
{  
    // -----  
    // declare services provided by HOST partition  
    // -----  
    server HOST <- FPGA;  
    {  
        method PrintMessage(in MSG_CLASS, out STATUS);  
        method PrintStat(in MSG_CLASS);  
        method PrintEvent(in MSG_CLASS);  
    };  
  
    // -----  
    // declare services provided by FPGA partition  
    // -----  
    server FPGA <- Host;  
    {  
        method Start();  
        method Stop();  
        method Step(in UINT);  
    };  
};
```

Questions?

Thank you

