

Architecture, Design and Implementation of Embedded Systems for Real-Time Applications

CpE-450 Spring 05

Class 4

Bruce McNair

bmcnair@stevens.edu

Syllabus - by week

1. Introduction

- Definition of embedded system
- Constraints on embedded vs. standalone systems
- Concept of real-time design
- Time scales for real-time systems
- Applications

2. Hardware/software functional partitioning

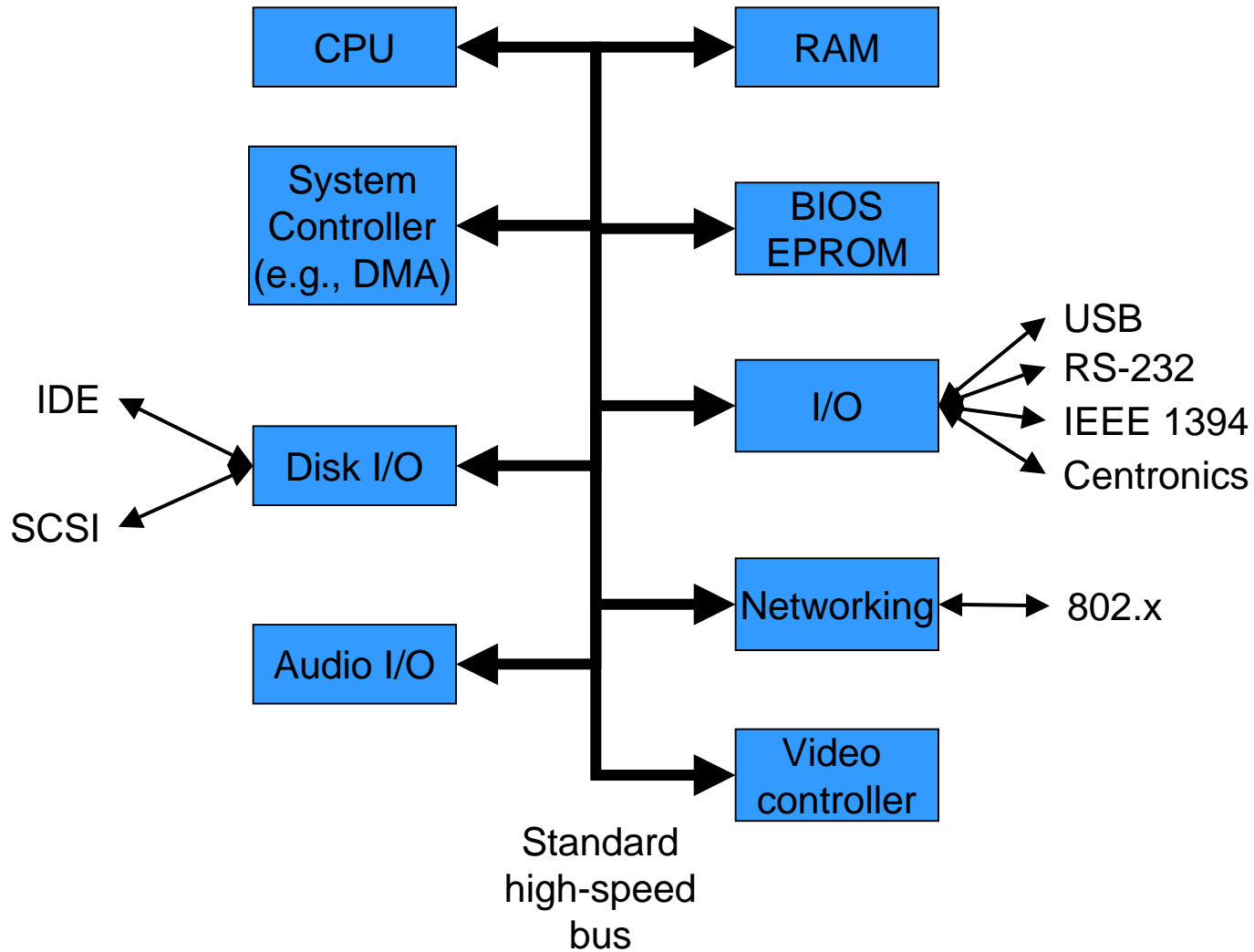
- Relevant hardware technologies:
 - Discrete logic
 - CPLDs, FPGAs, ASICs
- Software environments
 - HLL vs. assembly coding
 - DSP vs. general purpose vs. RISC

3. Development environments, course project definition

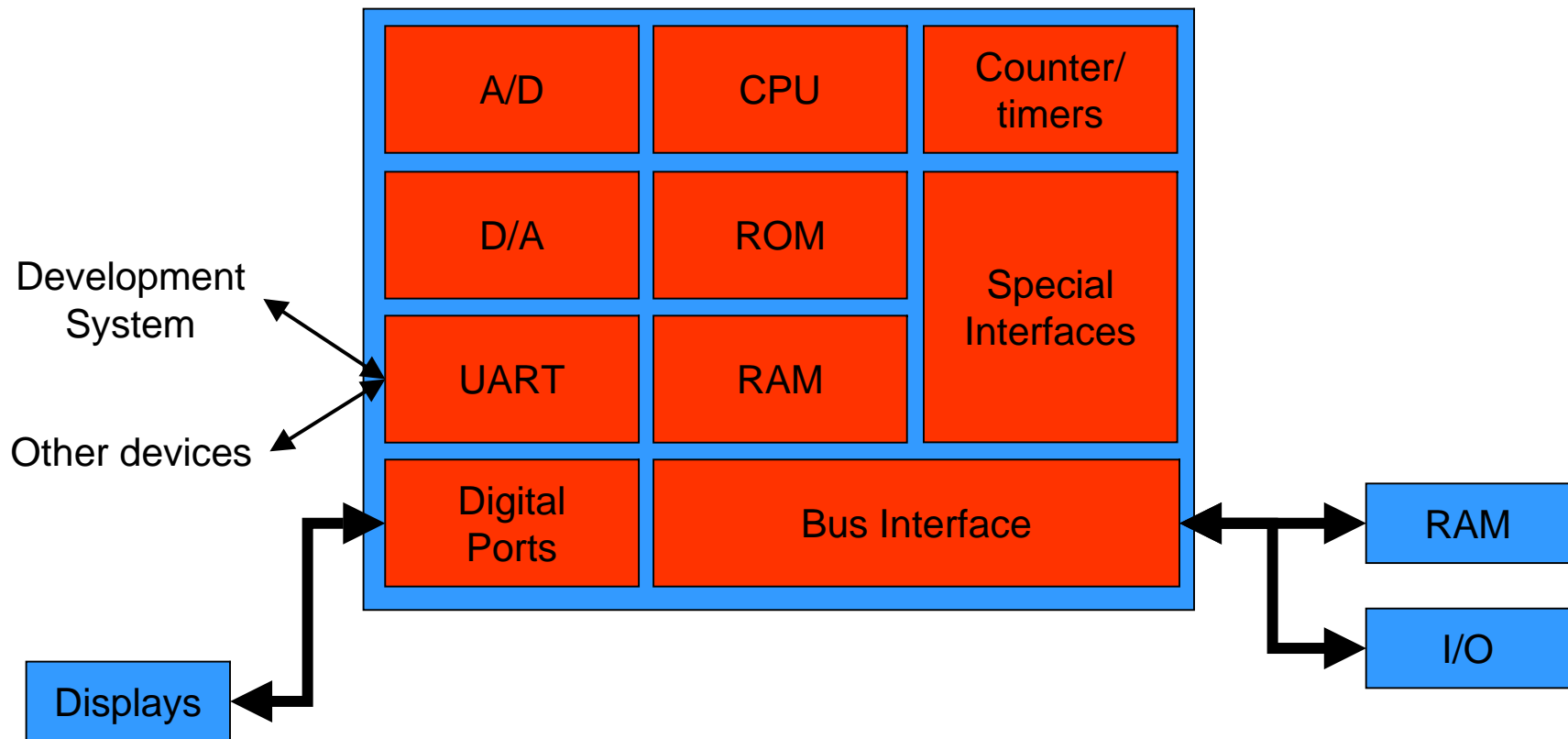
4. System architectures

5. Pipelining, interrupt service routines

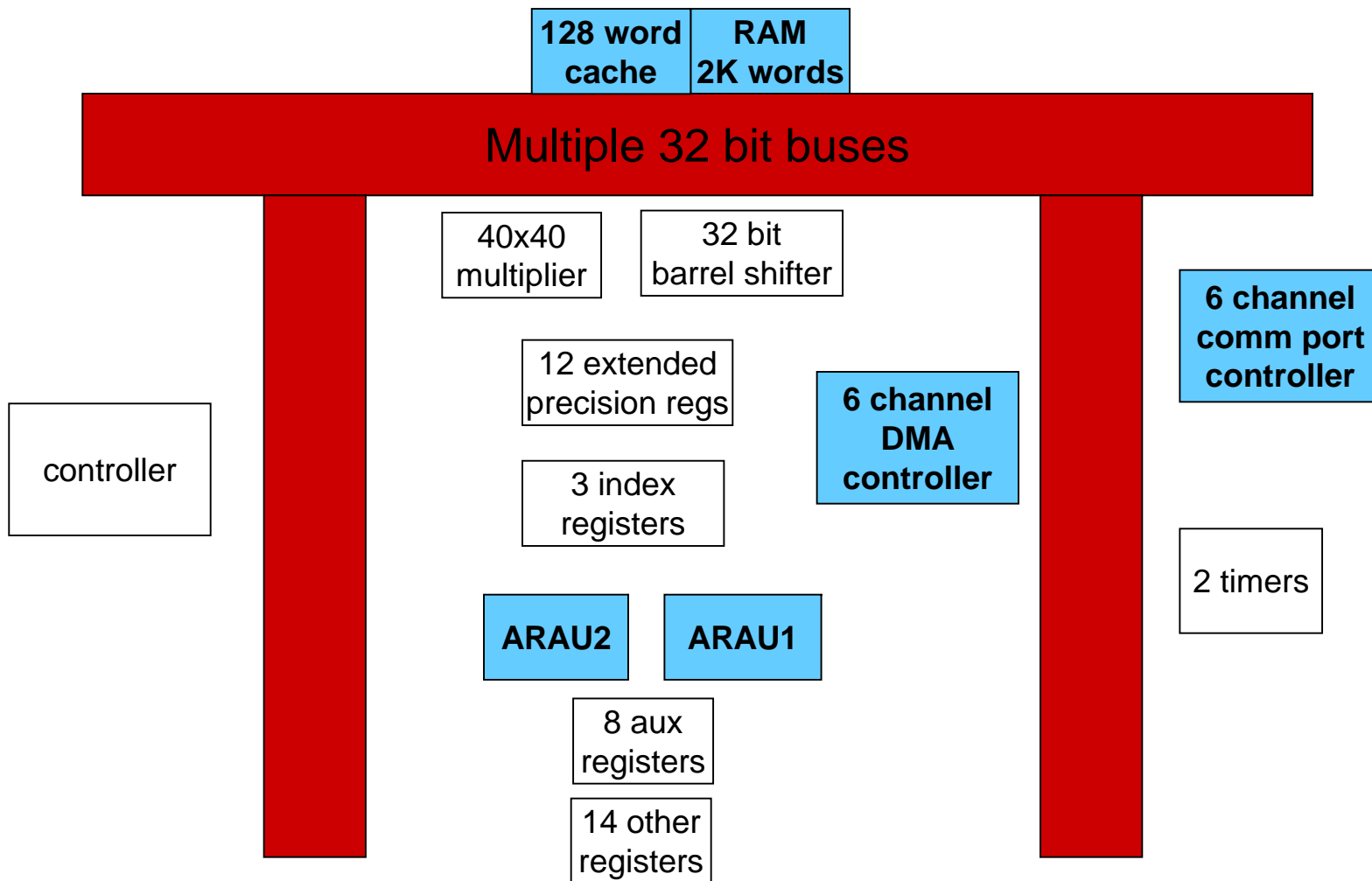
Review of Computer Architecture



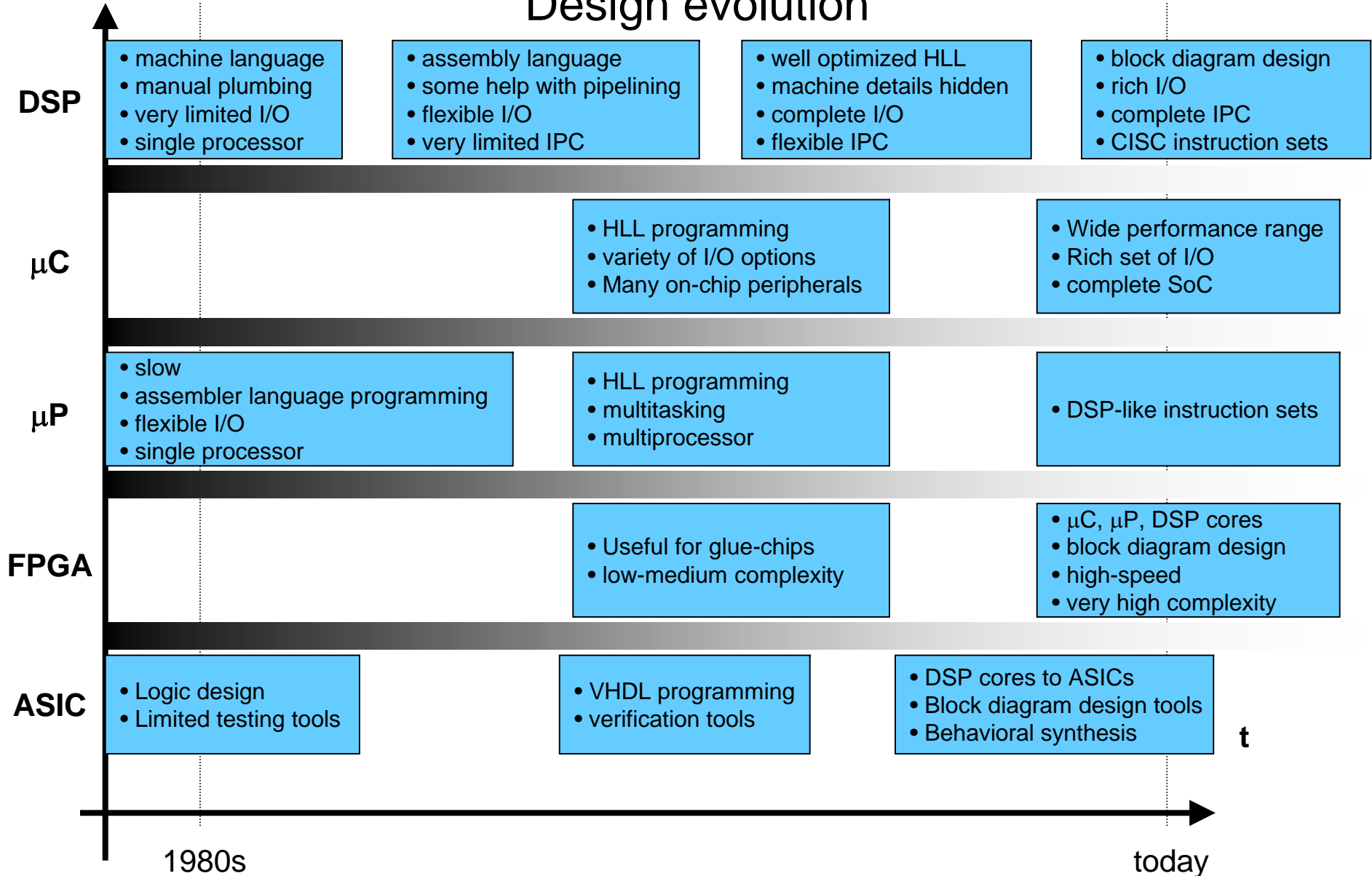
Embedded System Architecture



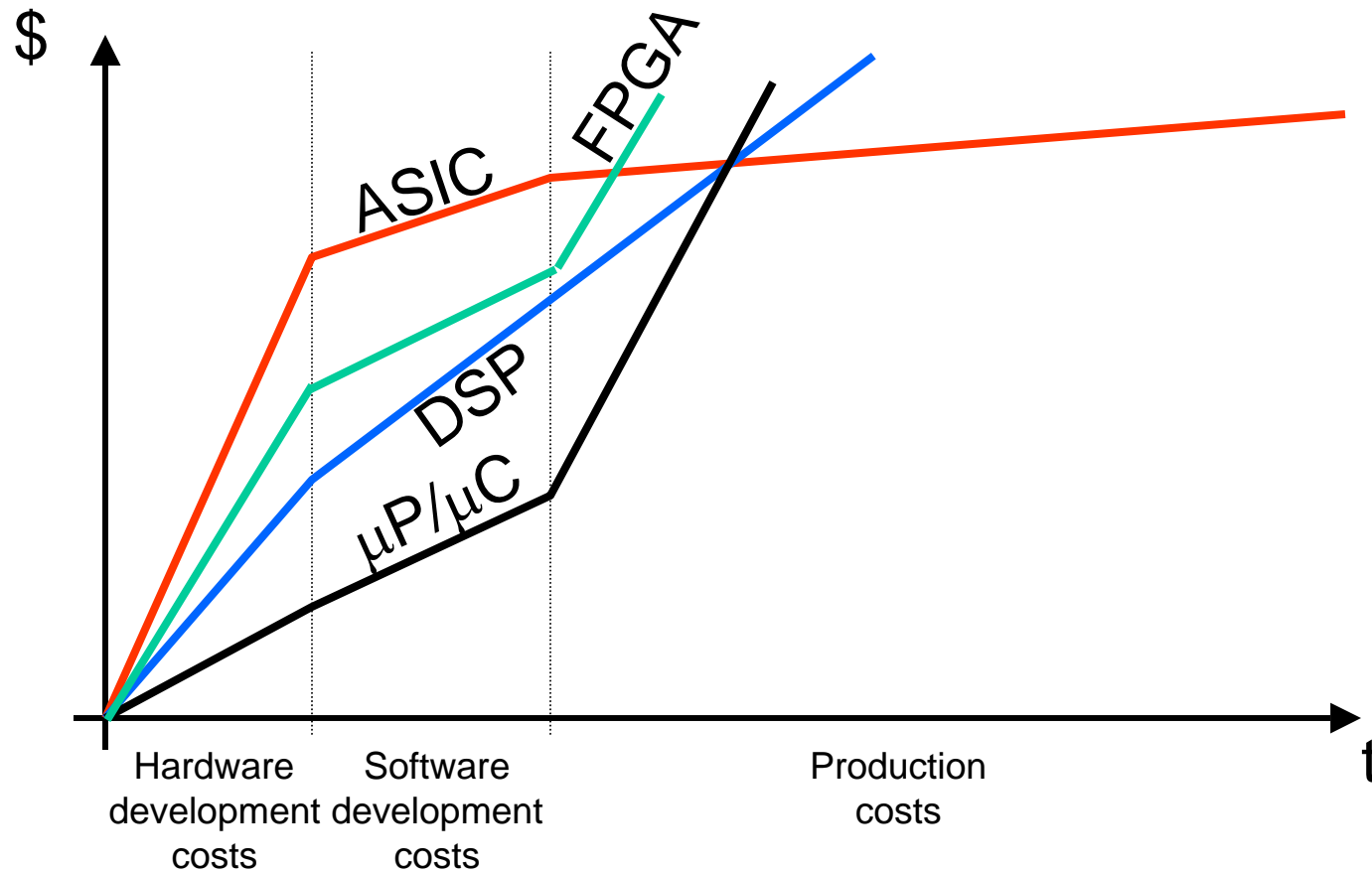
DSPs – the Ultimate Embedded Processors: TMS320C40 Architecture



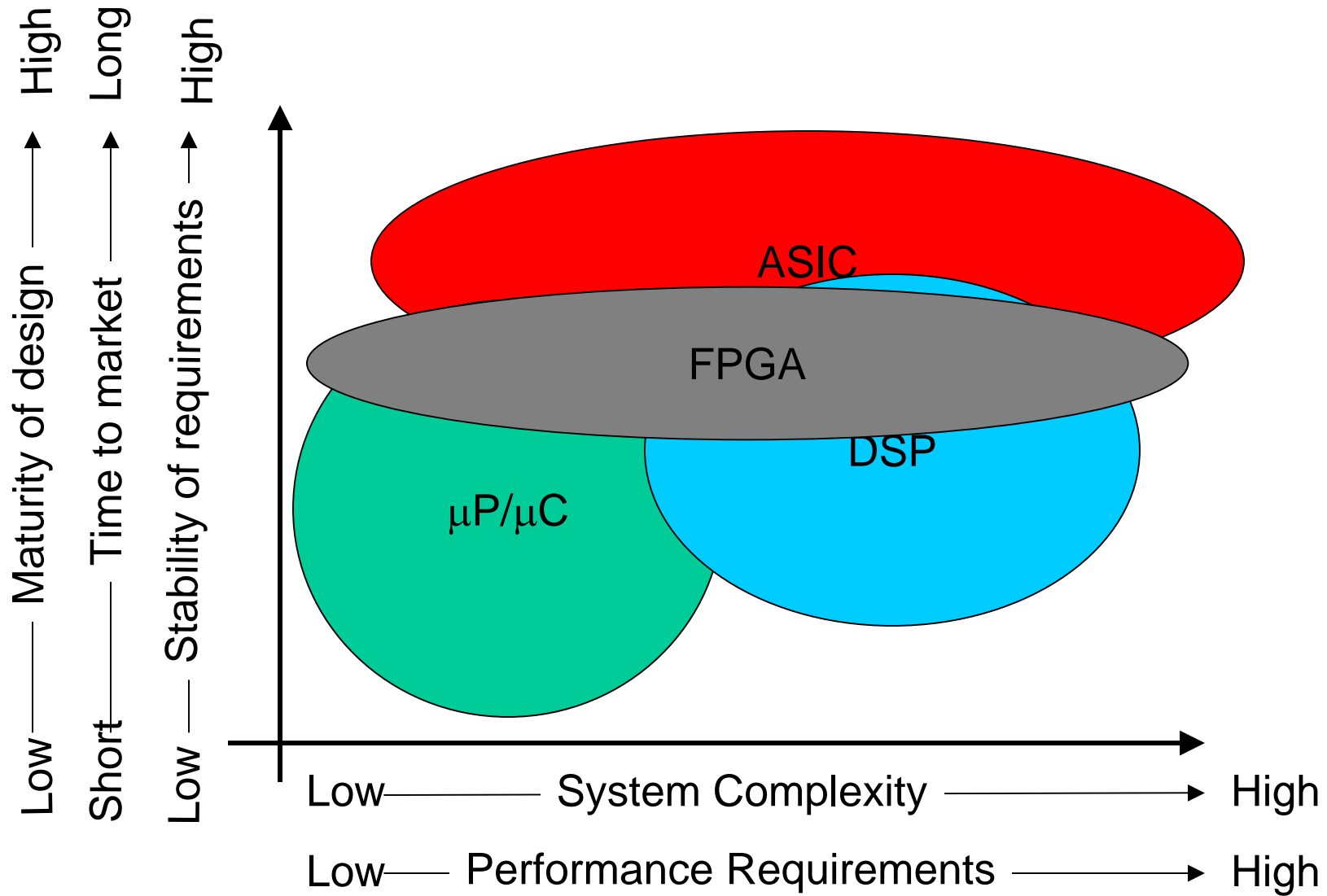
Design evolution



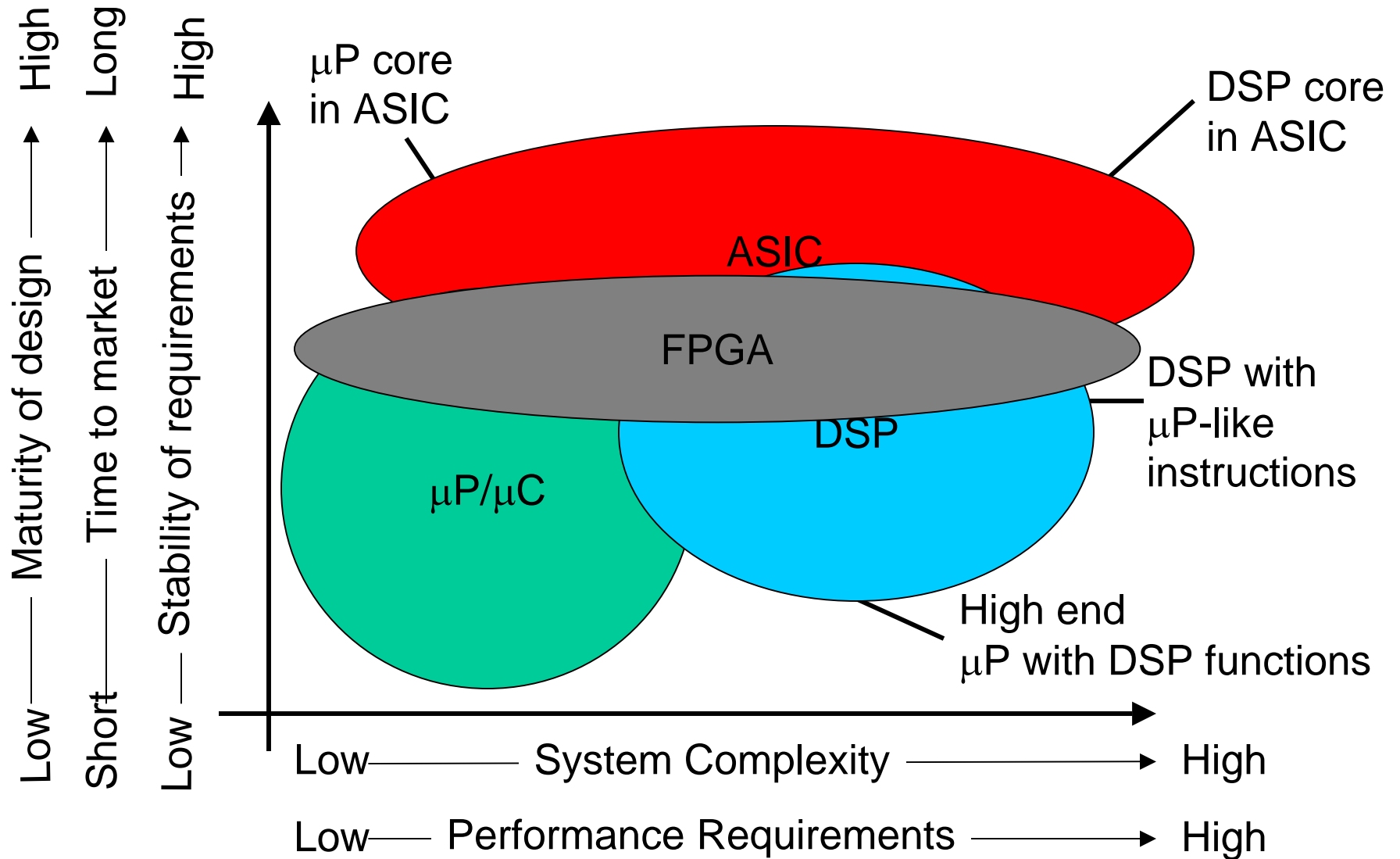
DSP versus ASIC versus μ P/ μ C



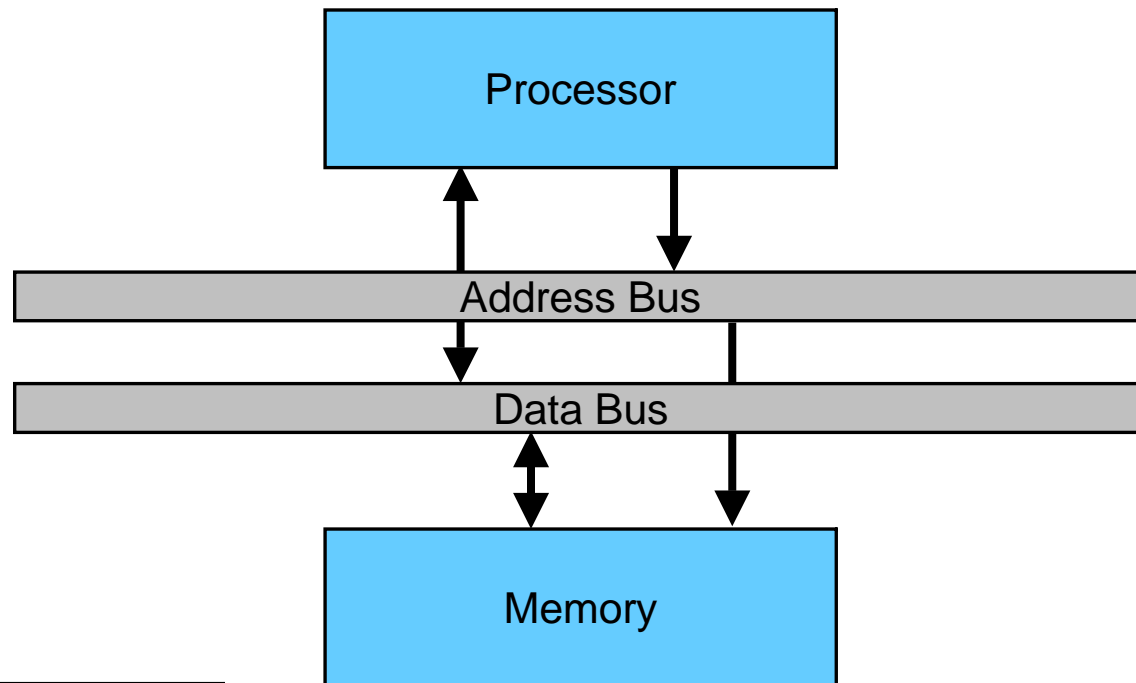
DSP vs. ASIC vs. FPGA vs. μ P/ μ C



DSP vs. ASIC vs. FPGA vs. μ P/ μ C



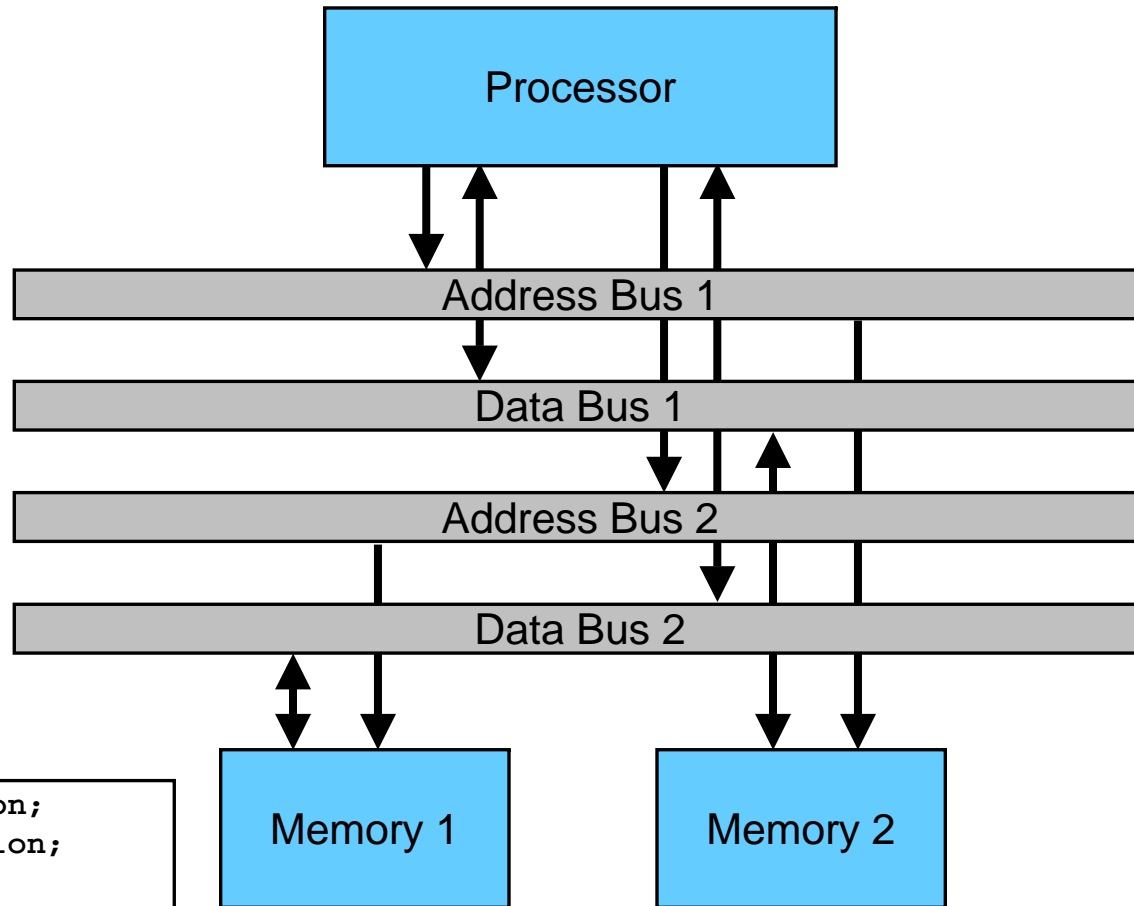
Von Neumann Architecture



```
fetch instruction;  
decode instruction;  
fetch data1;  
fetch data2;  
multiply/accumulate;  
write data;
```

Single buses create bottleneck

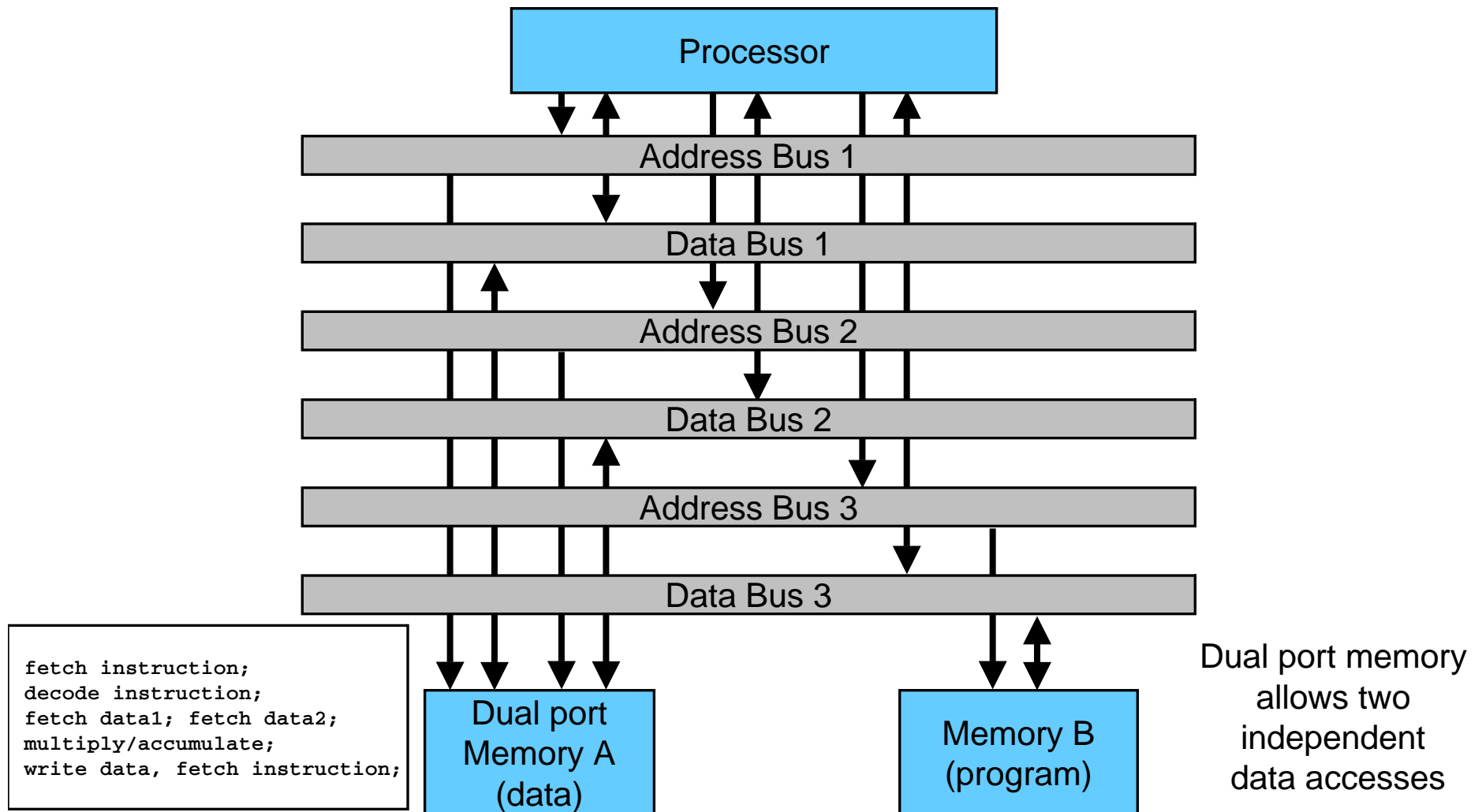
Harvard Architecture



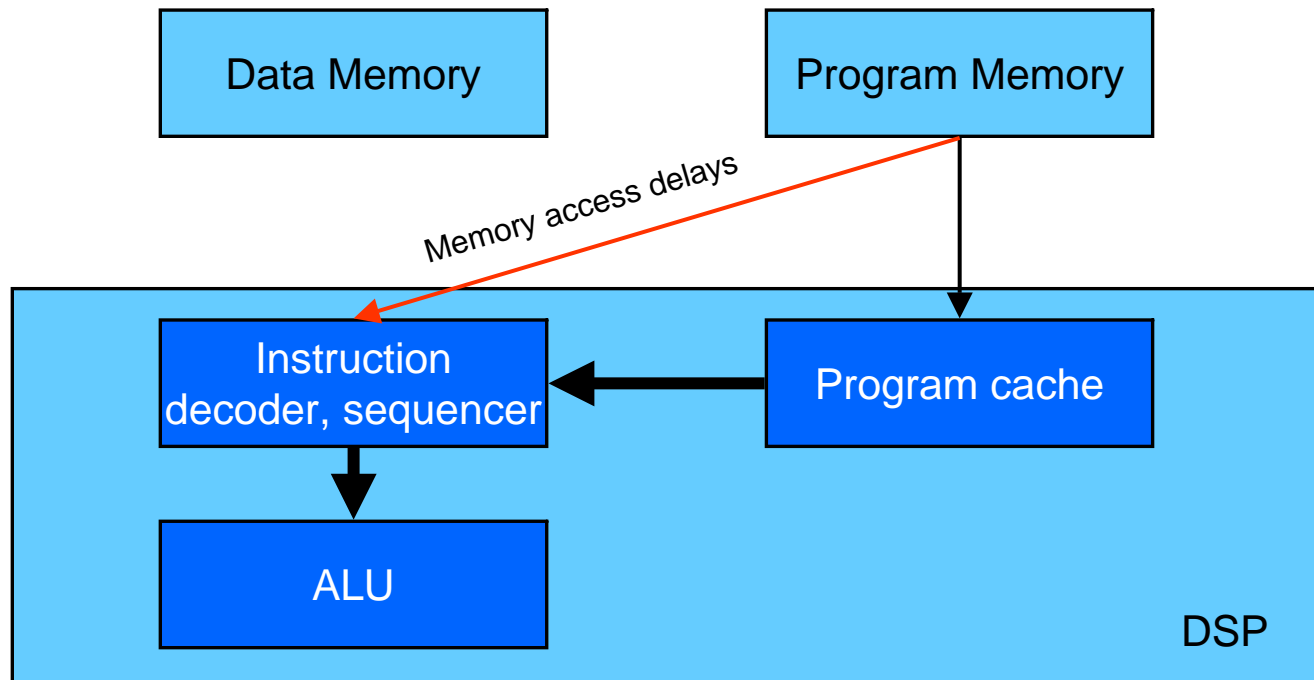
```
fetch instruction;  
decode instruction;  
fetch data1;  
fetch data2;  
multiply/accumulate;  
write data, fetch instr;
```

Multiple buses allow (limited) parallel operations

Harvard Architecture with enhanced memory access

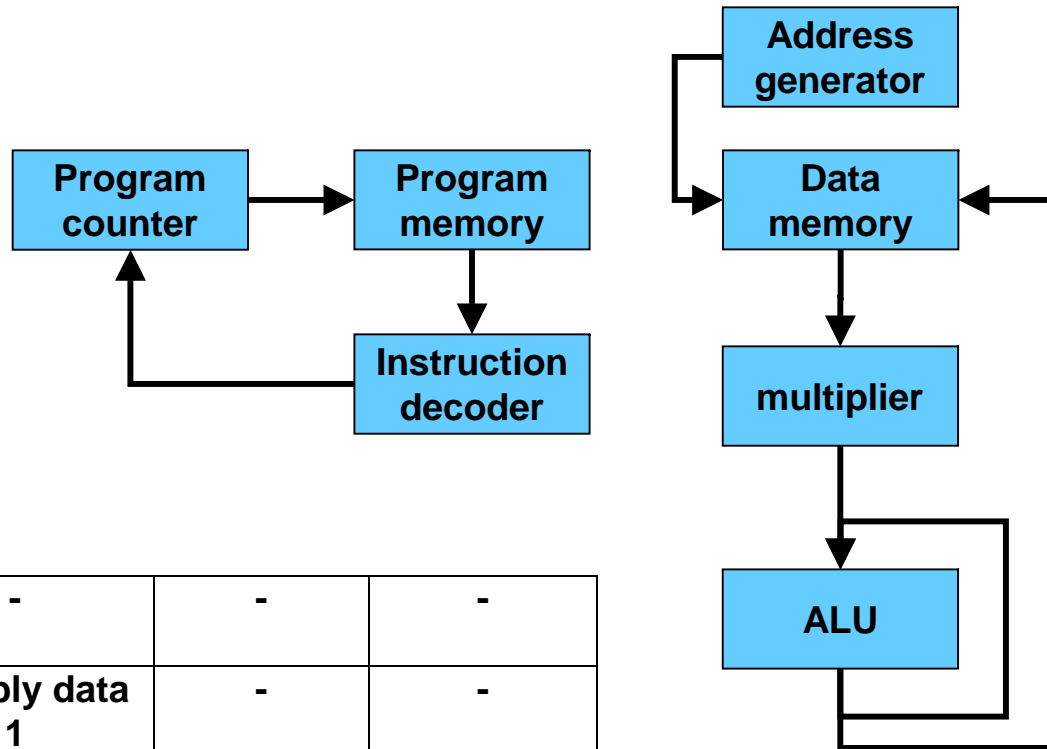


Caching



Most programs have the property of “locality”
- a small group of routines are exercised frequently
These instructions are candidates for “caching”
- storing on-chip for fast access

Pipelining instructions



Fetch data 1	-	-	-
Fetch data 2	Multiply data 1	-	-
Fetch data 3	Multiply data 2	Add data 1	-
Fetch data 4	Multiply data 3	Add data 2	Store data 1