

An Initial Study of an Obsolescence-Resilient Computer System Architecture based on the Bare Machine Computing Paradigm

October 16, 2026

Authors:

Dr. Ramesh K. Karne rkarne@towson.edu dr.rkarne@gmail.com

Dr. Alexander L. Wijesinha awijesinha@towson.edu

Department of Computer and Information Sciences

Towson University

Towson, MD 21252, USA

Dr. Nirmala Soundararajan nsoundararajan@se.edu

Computer and Information Science

John Massey School of Business

Southeastern Oklahoma State University

Durant 74701, OK

Website:

<https://www.karnebmc.com>

Agenda

- Motivation
- Obsolescence
- Computing Paradigms
- Engineering Principles
- UCU Architecture
 - Architecture
 - Approach
 - Mapping Results
- Design Issues
- Conclusion

Motivation

- Obsolescence (mostly business and profits driven, also technology driven)
- Security (never ending problem, patching approach)

Obsolescence

- Planned
- Design
- Appearance
- People skills
- Frequent
- Ripple effects
- Business and profit motives
- Consistent revenue driven
- Technology

21-Dimensional Obsolescence Space

Memory Chips 12
Memory Technologies 11

Releases-Versions 21

Storage Devices 14
Storage Technologies 13

Applications 06

Operating Systems 05

Executable Modules 04

Instruction Set Architectures 03

Processor Chips 02

Processor Technologies (Microns) 01

End-user Products 16
End-user Assemblies 15

Standards 10

I/O Devices/Drivers 18
I/O Technologies 17

Programming Languages 07

User Interfaces 08

Protocols 09

Software Vendors 19

Hardware Vendors 20

Computing Paradigms

- Conventional
- Bare Machine Computing (BMC)

Conventional

- Evolutionary
- OS or some middleware based
- Prone to obsolescence
- Heterogeneity driven
- Layered
- Rapid Obsolescence

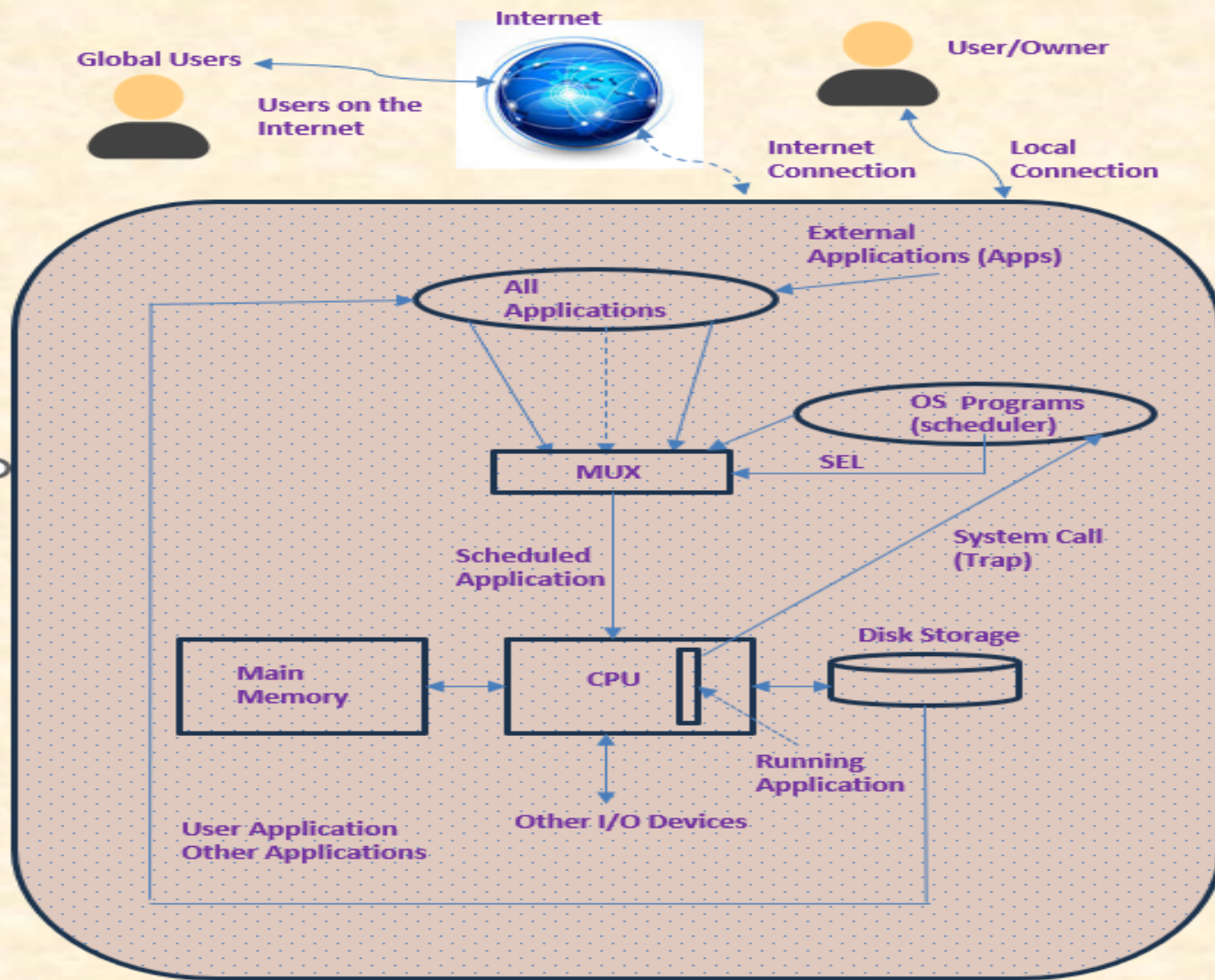
Bare Machine Computing (BMC)

- Started in 1995
- Motivated by complexity and obsolescence
- Frustrated by too many environments
- Initially funded by NSF grants
- Over 70 publications and 27 doctoral dissertations
- Started from scratch and built all BMC code and over a dozen complex applications over the years

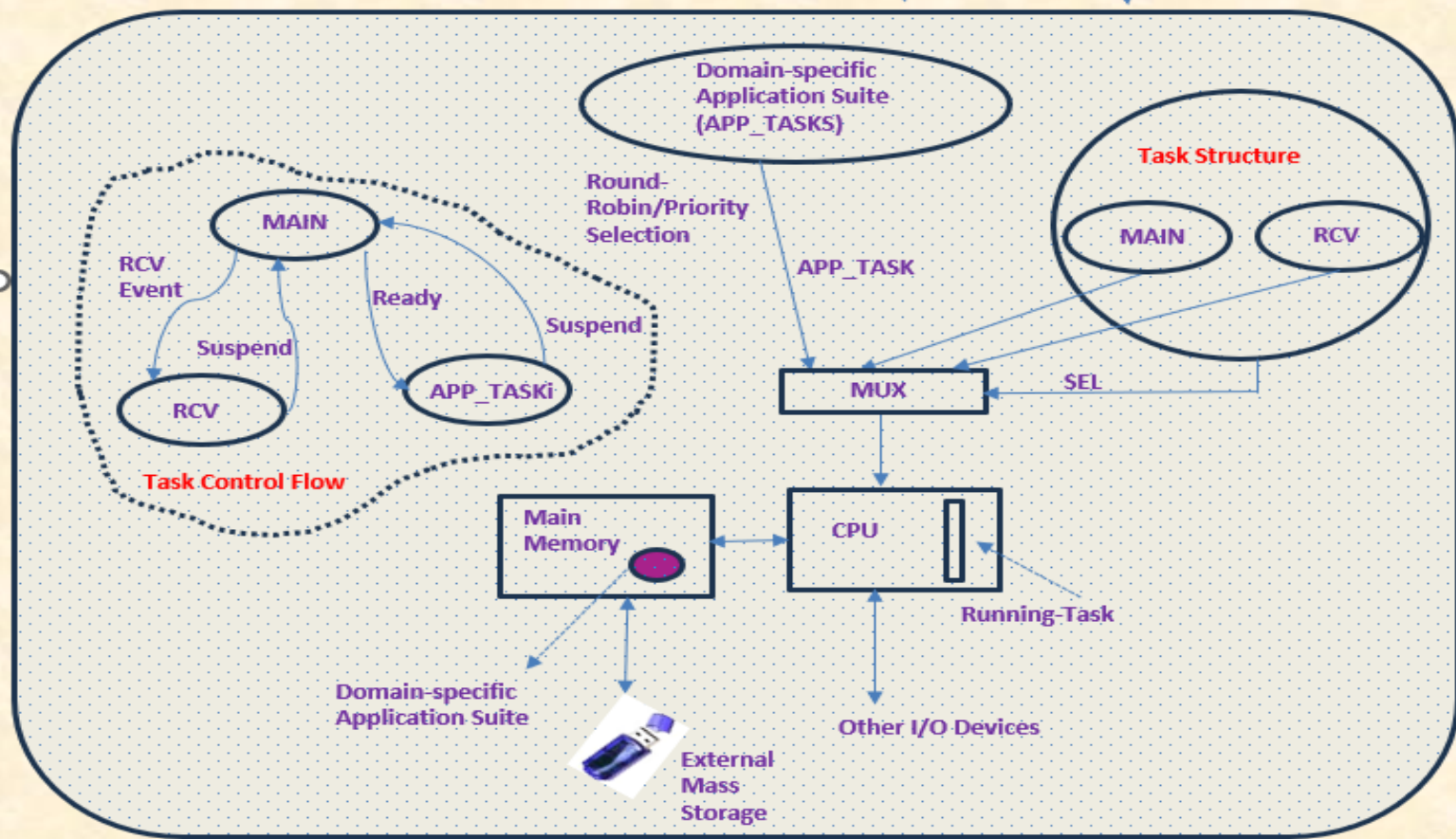
Bare Machine Computing Continued

- A computing device is bare
- No OS or any persistent storage (disk)
- No ownership
- Owner controlled
- Event driven
- Program time control flow
- Applicable to any computing device

Inner workings of a conventional computing device



Inner workings of a bare computing device



Engineering Principles

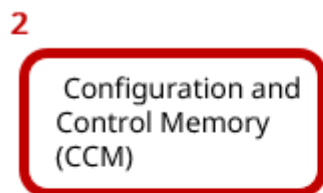
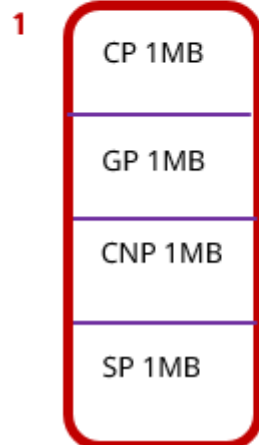
1. Universal Computing Unit (UCU)
2. Bare Computing Device (BCD)
3. BMC Paradigm
4. No OS or middleware
5. Direct Hardware Interfaces
6. Single Programming Language Base
7. Domain Specific Applications
8. Standard I/O interfaces
9. Standard Instruction Set Architecture
10. Object-oriented Methodology

Universal Computing Unit (UCU) Outline

1. A single processing unit
2. Contains 4 cores only
3. Embraces computing, communication, control and security applications
4. Bare Unit and follows BMC paradigm
5. Used for all applications
6. Small chip image
7. Massively produced and cheap
8. Domain-specific-application usage
9. Obsolescence-Resilient
10. Scalable for higher performance by using massively parallel approaches
11. Performance gain automatically achieved with technology evolution

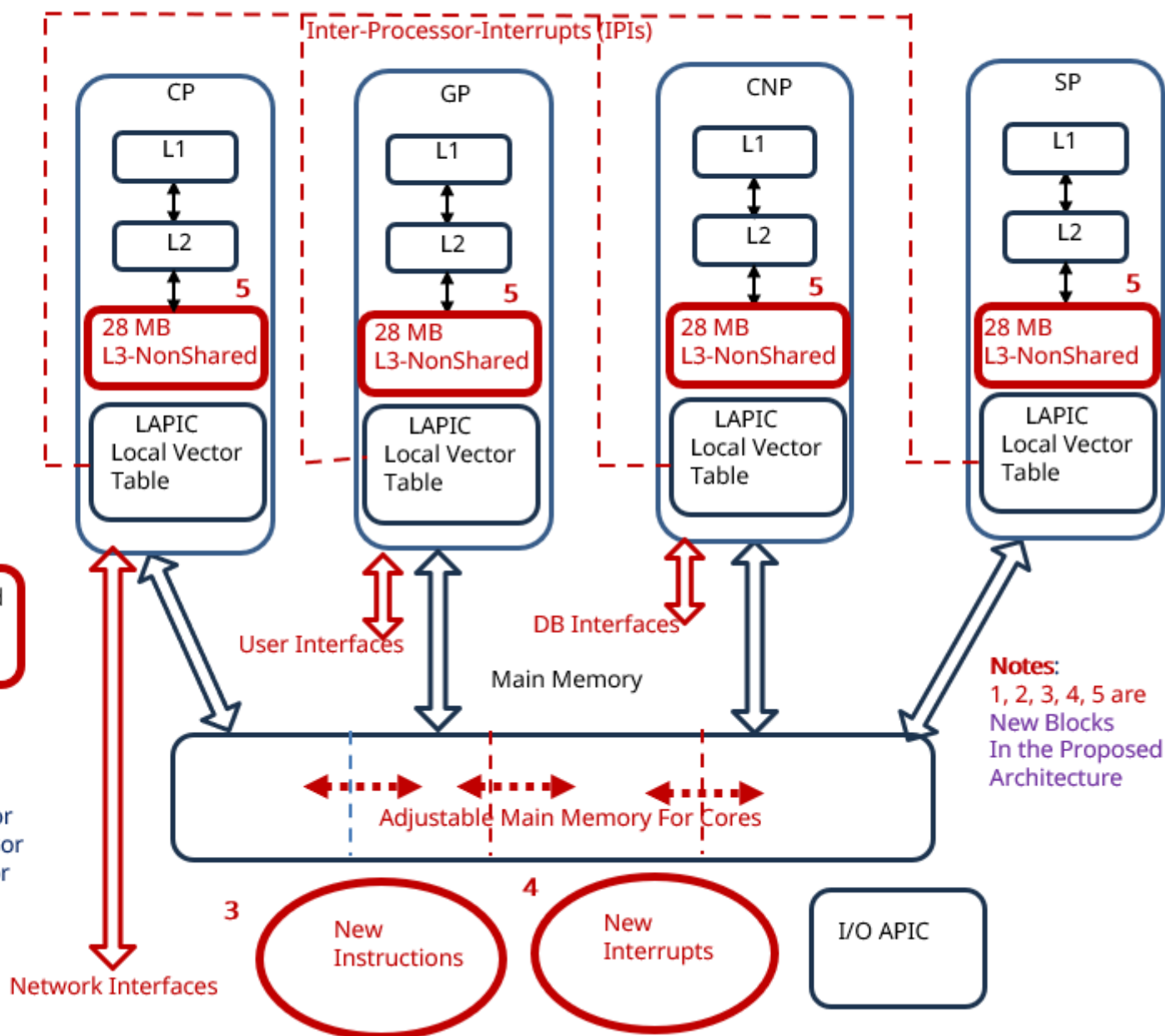
New Proposed Architecture

L3_Message Passing
Buffers in Each Core



Notes:

CP: Communication
Processor
GP: General Processor
CNP: Control Processor
SP: Security Processor
LAPIC: Local APIC



UCU Justification for building Obsolescence-resilient Systems

- Best way to justify this is to build a UCU and show that it accomplishes all computer and information systems and resilient to obsolescence
- As this is not possible with our resources and lack of collaboration with chip manufacturers, we show some initial study to demonstrate that our claims are valid
- Approach:
 - Consider a 2012 Intel i5-3570 CPU with four cores, 22-nm technology, and map to Intel i7-14700 CPU with 10-nm technology in 2024
 - Perform this mapping with existing data on technology mapping
 - Assume that the chip architecture stays the same in 12 years.

Mapping Results

Mapping results show the following data

1. The 24.3 billion transistors are leftover after mapping with the same architecture and functionality (Table 1); assuming the limitations of total power dissipation on the chip
2. L3 cache in i7-14700 was shared 33 MB. If the left over space is used to increase L3 cache, it can be 28MB of separate L3 cache for each core
3. By using 4MB of L3 cache in each core to communicate between them, then there will be 28 MB of separate cache in each core
4. The 4MB of communicating cache can be divided into 1MB each to hold communicating buffers

Observations of mapping results

Key Observations

1. As the BMC application images are small, 28-MB of L3 cache can be used as a main memory in many applications
2. A new wave of parallelism can emerge at L3 level
3. As the technology speeds increase, keeping the same architecture and functionality, more L3 can be realized thus reducing the main memory role
4. New instructions can be added to move data from L3 and Main memory as shown in the paper
5. Instead of increasing number of cores on the chip as it is done today, we can move more main memory on chip
6. Eventually, we can eliminate all main memory for most of the BMC applications
7. Using Intel i5-3570 CPU model, numerous variety of domain specific applications were built and demonstrated; it is possible to build other applications

Observations of mapping results continued...

- 8. Domain specific applications don't change; new functionality can be added in an object-oriented way to reduce obsolescence
- 9. The approach taken here illustrates that keeping the chip architecture same, new features can be added as needed with new technology
- 10. Currently, we have many chips, many instruction sets, and many CPU architectures; thus causing rapid obsolescence

Design Issues

There are many design issues as discussed in this paper, that have been studied and used in the BMC domain specific application suite development and deployment:

- Development process
- Memory management
- Software organization
- Simple task structure
- High level languages
- Scalability
- Cyber security
- Feasibility
- Resilience to obsolescence
- Drawbacks

Conclusion

- Presented a 21 dimensional obsolescence space
- The BMC paradigm can reduce obsolescence from 21 dimensions to 8
- Presented ten engineering principles that can help reducing obsolescence
- A new UCU architecture presented with 4 cores with comping, communication, control and security application domains
- Potential of L3 level parallelism and elimination of main memory concept presented
- Need a chip manufacturer to demonstrate the potential of the BMC and the proposed UCU.

Attributes	Intel i5-3570 CPU	Intel i7-14700 CPU	Parameters Change Over 12 years
Year	Q2, 2012	Q1, 2024	12-year range
Process	22 nm	10 nm	120% improvement
Generation	5th	14th	10-generations
Cores	4, Four Threads	20 cores, 28 threads	400% improvement
Clock Rate	3.4 GHZ	5.4 GHZ	59%
L1 Cache	32KB I Cache, 32KB D Cache	32K I-Cache + 48K D-Cache	16KB increase
L2 Cache	256KB each Core (1MB total)	8 p-cores have non shared 2 MB cache for each core; Other 12 e-cores have shared cache of 12MB (total 28MB)	27MB increase
L3 Cache	6 MB	33 MB	27 MB increase
Thermal Power Dissipation	77 Watts	65 – 219 Watts	
Socket (Contact Pad Layout is Confidential)	FCLGA1155	FCLGA1700	545 contact pads increase
Chip Die Size	160 mm ² [56]	257 mm ² [57]	97 mm ² increase
Total number of transistors on the chip	1.4 billion [56]	25.7 billion [58] See the following calculation: 100 million transistors per mm ² (257*100*10 ⁶) = ...	(25.7 – 1.4) = 24.3 billion leftover transistors

Table 2. TPD for normal and turbo power limits for i7-14700

Logic Function	Normal TPD	Turbo TPD
P-cores (8)	32.5 W	109.5 W
E-cores (12)	13 W	43.8 W
L3 Cache 33 MB shared	9.75 W	32.9 W
Interfaces Group	6.5 W	21.9 W
Other-Miscellaneous	3.25 W	11 W
TOTAL TPD	65 W	219 W

Table 3. TPD for normal and turbo power limits in proposed architecture

Logic Function	Normal TPD	Turbo TPD	Comment
P-cores (4)	16.25 W	54.7 W	½ of Table 2
E-cores (12)	0	0	No E-cores
Total L3 Cache 128 MB	37.8 W ((128 * 9.75)/33)	127.6 W (128*32.9/33)	New values derived proportionate to old data values
Interfaces Group	6.5 W	21.9 W	Same as in Table 2
Other Miscellaneous	3.25 W	11 W	Same as in Table 2
TOTAL TPD for 128 MB L3 Cache	63.8 W	215.2 W	TPD Within Limits of (65W, 219W)

Table 4. Comparison of Current and Proposed Systems using UCU

Dimension/Attribute	Current Systems	Proposed UCU Systems
1. Chip Technologies	Many (n)	1 (choose the best)
2. Chips	Many (n)	1 (choose the best)
3. ISA	Many (n)	1
4. Executables	Many (n)	1
5. OS	Many (n)	0
6. Applications	Many (n)	DSAs (m) *
7. Programming Languages	Many (n)	1
8. User Interfaces	Many (n)	1 (USB Types)
9. Memory Technologies	Many (n)	1 (choose the best)
10. Memory Chips	Many (n)	1 (choose the best)
11. Storage Technologies	Many (n)	1 (USB)
12. Storage Devices and Drivers	Many (n)	1 (USB)
13. I/O Technologies	Many (n)	1 (USB)
14. I/O Devices and Drivers	Many (n)	1 (USB)
15. End User Assemblies	Many (n)	Many (n) *
16. End User Products	Many (n)	Based on Domains (n) *
17. Protocols	Many (n)	Limited (m) *
18. Standards	Many (n)	Limited (m) *
19. Software Vendors	Many (n)	Domain Specific (n) *
20. Hardware Vendors	Many (n)	Limited (m) *
21. Releases and Versions	Often, Many	Less Often (m) *

