Processor Architecture and the Importance of the Storage Hierarchy

Ray Wicks
ATS at the Washington Systems Center
561-236-5846
RayWicks@us.ibm.com

Abstract

Processor architecture and the importance of the storage hierarchy
This session quickly covers the processor architecture and the role of the storage hierarchy in performance and capacity planning. This session provides an overview of the roles of the I-unit, E-unit, caching levels, and RAM. The emphasis is focused upon the movement of data and programs in the storage hierarchy versus the measure of performance. The measures seen are the cycles per instruction and the relative nest intensity. Question to be answered: How can two machines with the same memory size and cycle time perform with such different capacities?

Bibliography

Ray has spent most of his career at IBM in the performance analysis and capacity planning end of the business in Poughkeepsie, London, and now at the Washington Systems Center. He is the major contributor to IBM's internal PA & CP tool zCP3000. This tool is used extensively by the IBM services and technical support staff world wide to analyze existing zSeries configurations (Processor, storage, and I/O) and make projections for capacity expectations.

Ray has given classes and lectures worldwide. He was a visiting scholar at the University of Maryland where he taught part time at the Honors College.

He won the prestigious Computer Measurement Group's A.A. Michelson award in 2000..

Trade Marks, Copyrights & Stuff

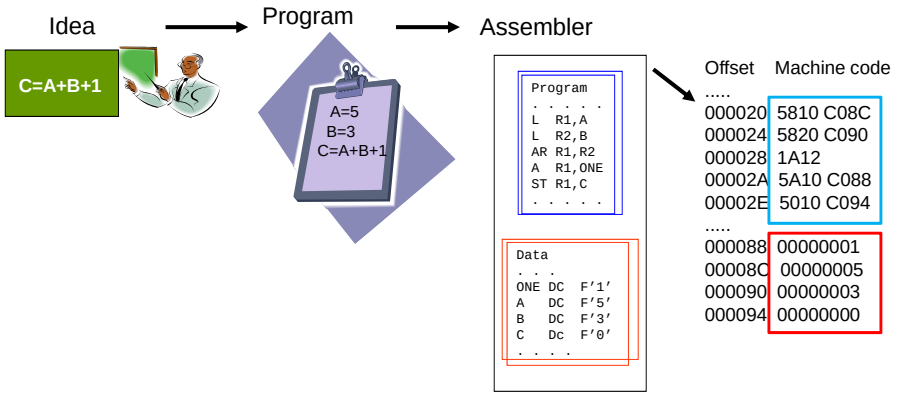
Many terms are trademarks of different companies and are owned by them.

- **On foils that appear in this presentation are not in the handout. This is to prevent you from looking ahead and spoiling my jokes and surprises. Also foils added after I made handouts.**

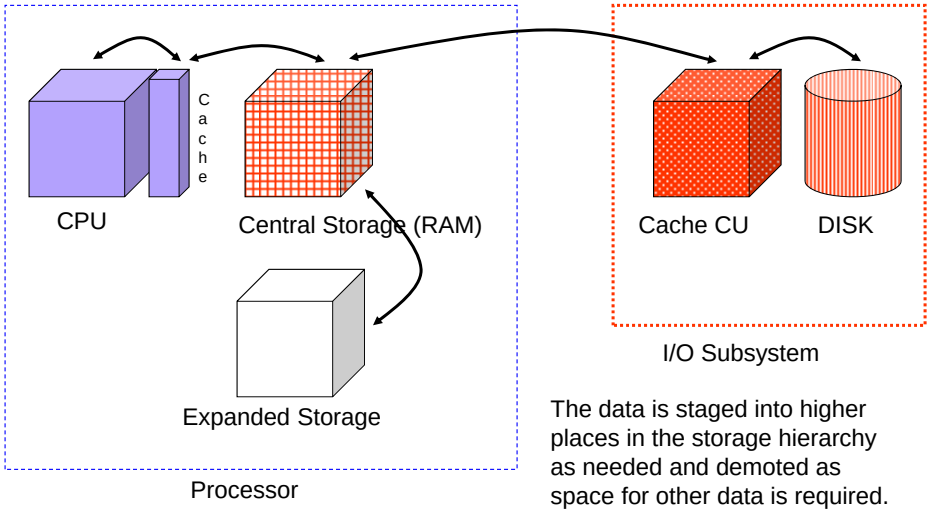
The Processor

CEC = Central Electronic Complex
CPC = Central Processor Complex

Programming

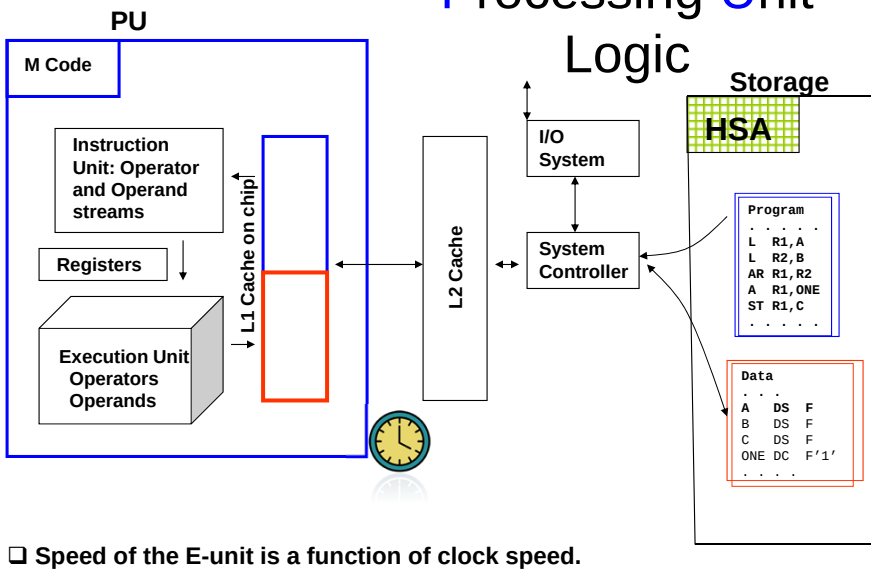


Computer Organization



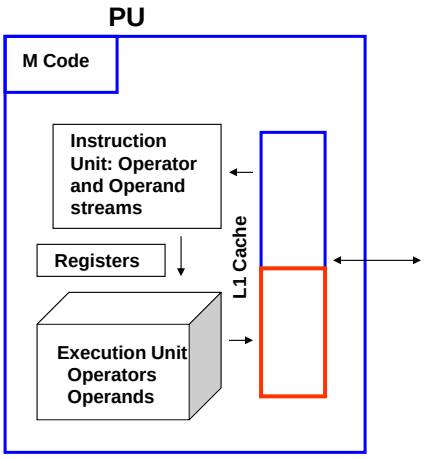
The data is staged into higher places in the storage hierarchy as needed and demoted as space for other data is required. Where do you want your data?

Processing Unit Logic



- ❑ **Speed of the E-unit is a function of clock speed.**
- ❑ **Effective E-Unit speed is a function of the data location.**

Processing Unit Logic

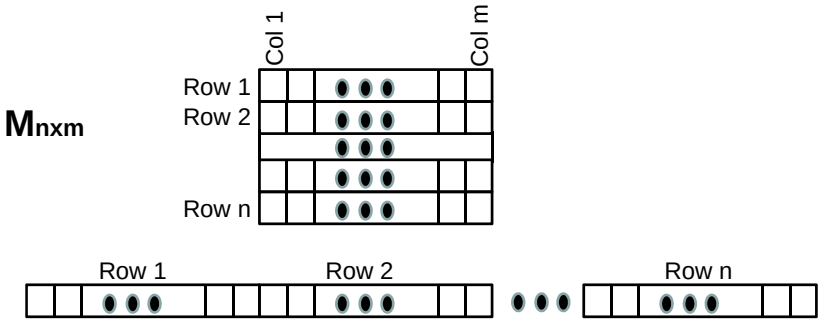


Roughly:

1. **Instruction fetch (from??) cycle**
2. **Instruction decode/Register fetch cycle**
3. **Data fetch cycle**
4. **Execution/Effective address cycle**
5. **Memory access cycle**
6. **Write-back cycle**

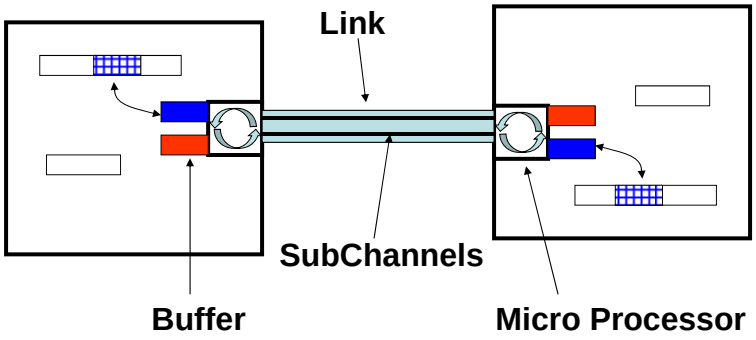
- **Speed of the I & E-unit is a function of clock speed.**
- **Effective E-Unit speed is a function of the architecture & data location.**

Matrix Processing & Memory



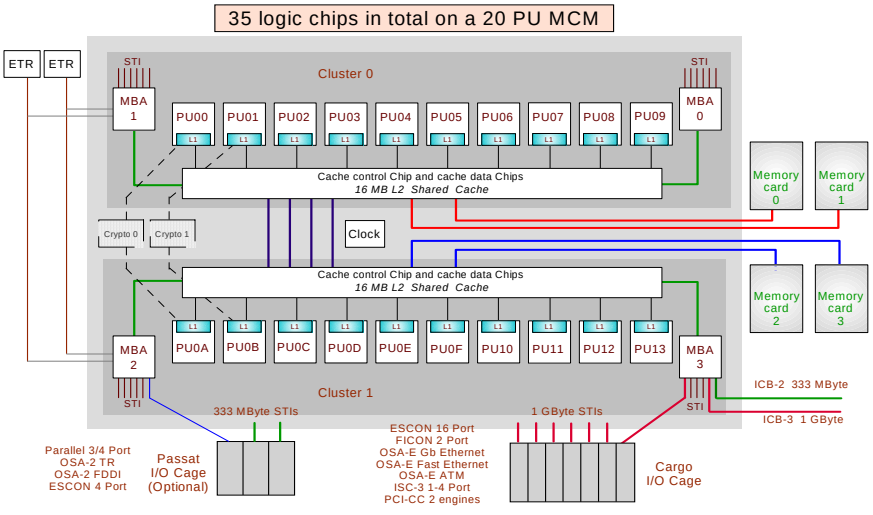

```
DO I=1 to n                    DO J=1 to m
DO J= 1 to m          or       DO I= 1 to n
M[I;j] = f(x,t)                M[I;j] = f(x,t)
End;                           End;
End;                           End;
```

Connections



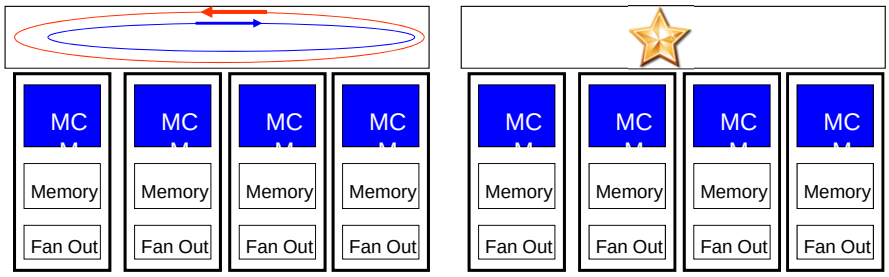
Every line connecting two boxes in a diagram implies micro processors on each end to do the talking? (What happens if they speak different languages?) Data is moved from a buffer to micro processor buffer onto link into m-processor buffer into storage buffer.

Z900 Structure



Ref. SG24-5975

Book Structure

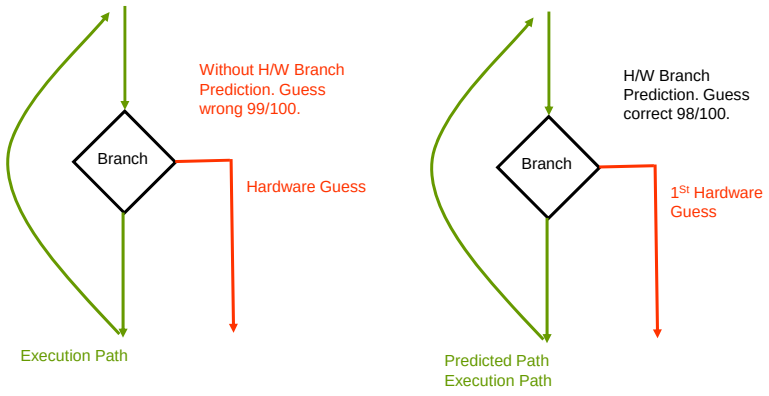


Architecture, Implementation & Packaging

```
L   R1,A
L   R2,B
AR  R1,R2
A   R1,ONE
ST  R1,C
```

- Instructions executed in sequence?
- Data
  - Prefetched?
  - In Cache?
- Cycles per instruction?
- Branch?
- Branch Prediction?
- Parallelism?
- Distances? (Packaging density)
- Many PUs Interference?
- Micro Code capability?
- Etc.

Branch Prediction



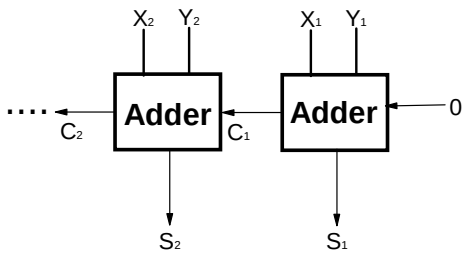
When hardware guesses, it prefetches instructions & operands, decodes and pre-executes. If guess is not correct, it discards the guess.

Cycles Per Instruction

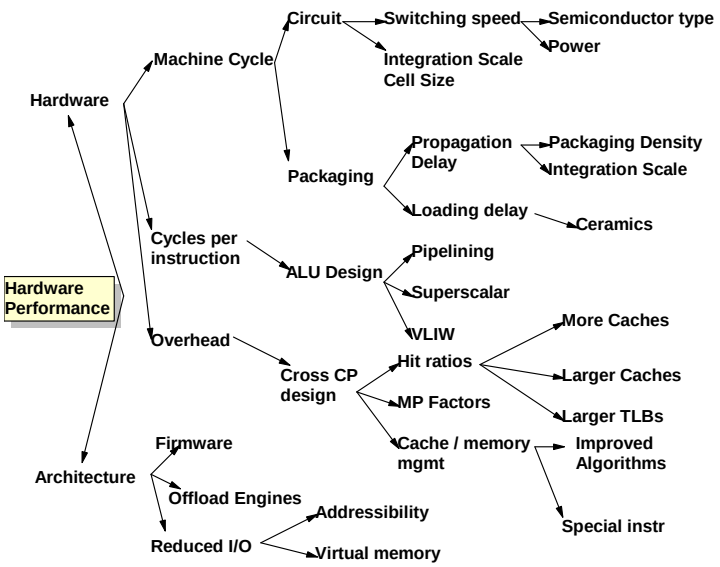
+	0	1
0	00	01
1	01	10

+	0	1
0	0	1
1	1	0,C1

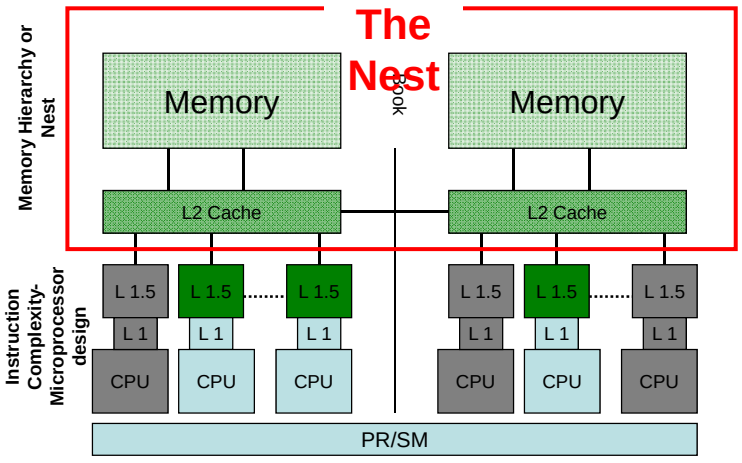
$$[x_n,\ldots..x_2,x_1]$$
$$+ [y_n,\ldots..y_2,y_1]$$



Etc.



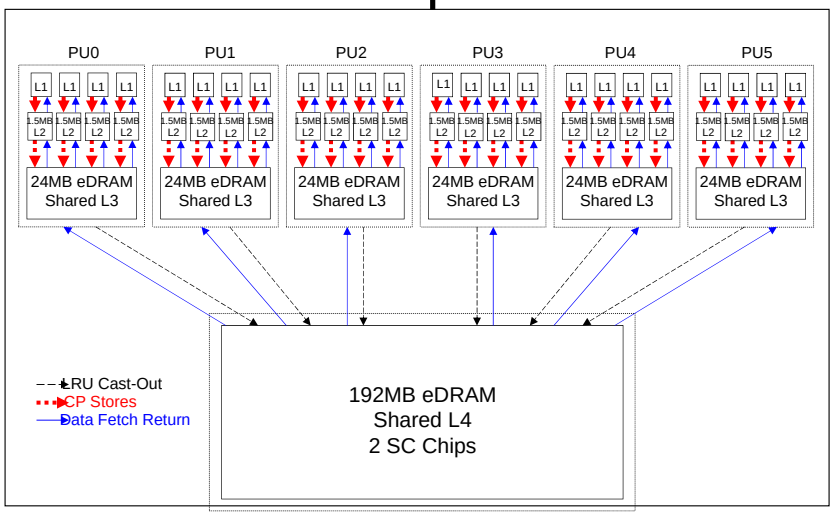
Z10 Memory – a simple view



Reference: John Burg's presentation at SHARE 3/3/2011

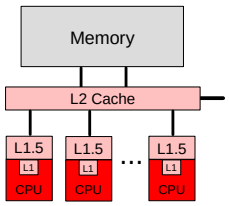
http://www/ibm.com/support/techdocs/atsmastr.nsf/Webindex/TC000066

zEnterprise

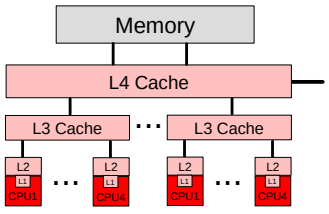


z196 versus z10 hardware comparison

- z10 EC
  - CPU
    - 4.4 GHz
  - Caches
    - L1 private 64k i, 128k d
    - L1.5 private 3 MB
    - L2 shared 48 MB / book
    - book interconnect: star



- z196
  - CPU
    - 5.2 GHz
    - Out-Of-Order execution
  - Caches
    - L1 private 64k i, 128k d
    - L2 private 1.5 MB
    - L3 shared 24 MB / chip
    - L4 shared 192 MB / book
    - book interconnect: star

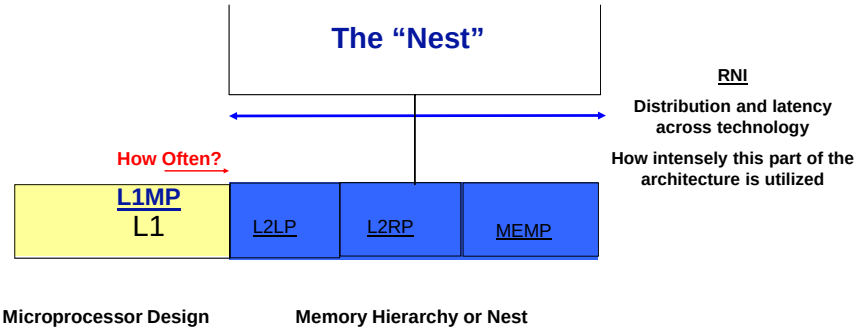


20

Introducing the new Relative Nest Intensity (RNI) metric

(SMF 113 Data)

Relative Nest Intensity

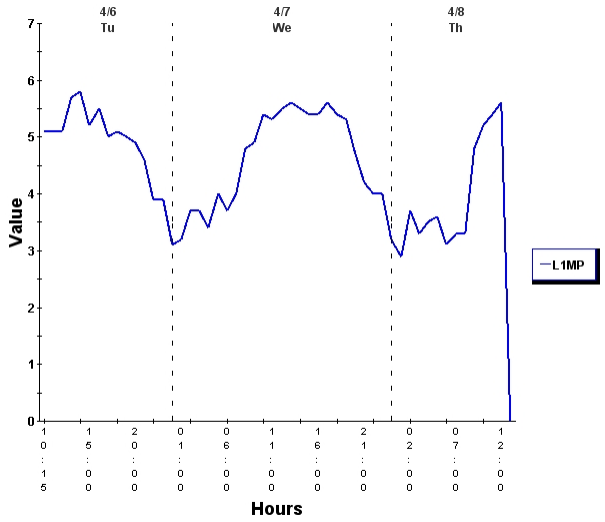


Definitions

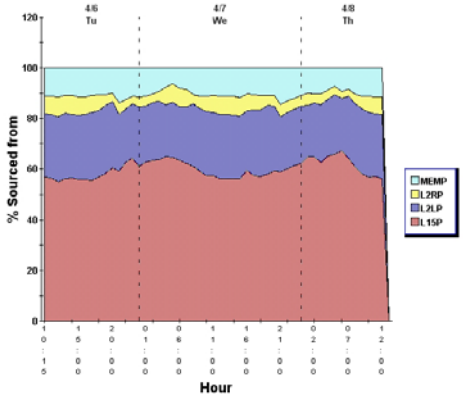
CPI – Cycles per Instruction
PRB STATE - % Problem State
L1MP – Level 1 Miss Per 100 instructions
L15P / L2P – % sourced from L1.5 or L2 cache
L2LP – % sourced from Level 2 (or L4) Local cache (on same book)
L2RP – % sourced from Level 2 (or L4) Remote cache (on different book)
L3P – % sourced from L3 cache
MEMP - % sourced from Memory
Rel Nest Intensity – Relative Nest Intensity

(Ref: **CPU MF – Formulas and Updates by John Burg)**

Level 1 (L1) Miss Percent



If not from L1, from Where? (SMF 113 Data)



Here's the plot of percent sourcing from different levels of cache. As the sourcing moves from the highest level of cache (percent=L15P) to the slowest memory source (percent=MEMP), the performance degrades. Level 1 cache is the fastest and closest to the processing unit.

The sourcing shown in the graph is for data not found in level 1 cache. You can check the level 1 cache miss % by graphing variable L1MP.

- L1.5P=%sourced from level 1.5 cache
- L2LP=%sourced from level 2 cache same book
- L2RP=%sourced from level 2 cache different book
- MEMP=%sourced from memory

Remember than as more and more of the instructions and data has to be fetched from more distance caches, the machine effectively runs slower.

sas | Enterprise Guide
The Power to Know.

2097-E54 Sample

1 With Variables: CPI
5 Variables: L1MP L15P L2LP L2RP MEMP

Simple Statistics

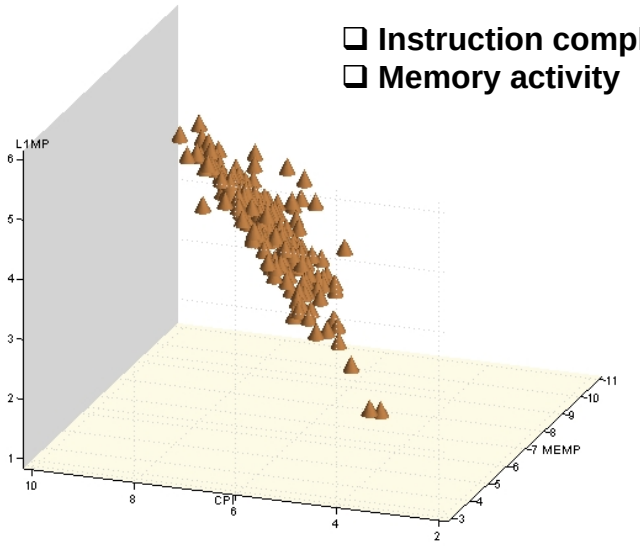
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum	Label
CPI	220	6.70909	0.82921	1476	3.90000	8.60000	CPI
L1MP	220	4.21091	0.61397	926.40000	1.90000	5.60000	L1MP
L15P	220	68.80273	2.73381	15137	62.50000	79.00000	L15P
L2LP	220	24.60364	1.92242	5413	17.70000	27.90000	L2LP
L2RP	220	0.09636	0.01876	21.20000	0	0.10000	L2RP
MEMP	220	6.51364	1.18811	1433	3.20000	10.40000	MEMP

Pearson Correlation Coefficients, N = 220
Prob > |r| under H0: Rho=0

	L1MP	L15P	L2LP	L2RP	MEMP
CPI	**0.79329**	-0.53761	0.41295	0.09019	**0.56472**
CPI	<.0001	<.0001	<.0001	0.1826	<.0001

Cycles per Instruction

- ❑ **Instruction complexity**
- ❑ **Memory activity**



Stepwise Regression

Stepwise Analysis
Table of Results for General Stepwise

L1MP entered.

	df	*SS*	*MS*	*F*	*Significance F*	***Rsquare***
Regression	1	94.76340452	94.76340452	370.1004889	7.13123E-49	**0.629315051**
Residual	218	55.81841366	0.256047769			
Total	219	150.5818182				

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	2.197521837	0.236981894	9.272952449	1.82816E-17	1.730452906	2.664590768
L1MP	1.071400255	0.055691885	19.23799597	7.13123E-49	0.96163681	1.181163699

MEMP entered.

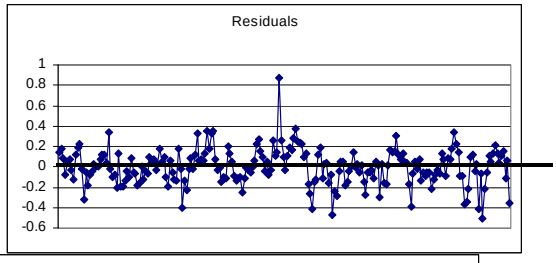
	df	*SS*	*MS*	*F*	*Significance F*	***Rsquare***
Regression	2	144.0333909	72.01669544	2386.469636	1.8304E-148	**0.956512497**
Residual	217	6.548427298	0.030177084			
Total	219	150.5818182				

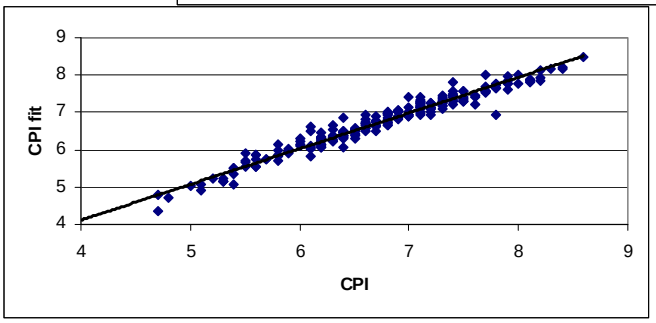
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-0.432760399	0.104193631	-4.153424666	4.70956E-05	-0.638121488	-0.22739931
L1MP	1.07847415	0.019120016	56.40550326	1.2808E-131	1.040789434	1.116158866
MEMP	0.399238545	0.009880522	40.40662553	6.1031E-103	0.379764469	0.418712621

Predict CPI in terms of Memory Activity

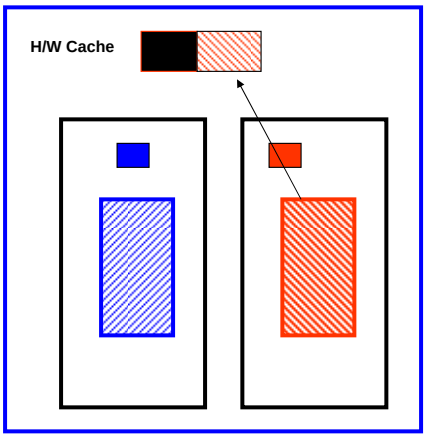
$CPIfit = a_0 + a_1 * L1MP + a_2 * MEMP$

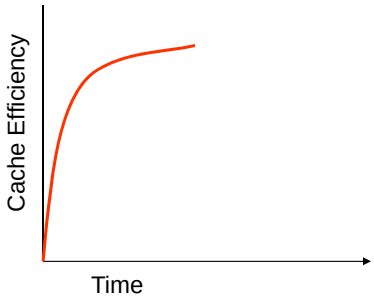
	Coefficients
Intercept	-0.43276
L1MP	1.078474
MEMP	0.399239





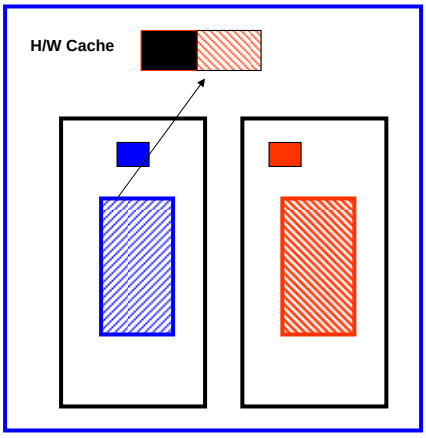
Task Elongation 1

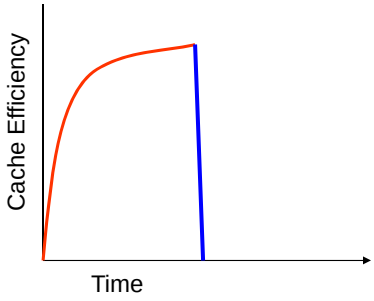




When Red partition is dispatched, its data is loaded into L1 cache. Cache efficiency improves over time.

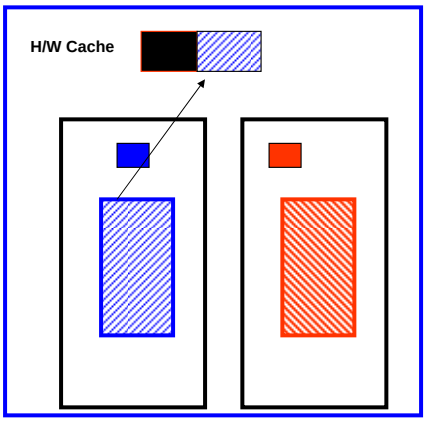
Task Elongation 2

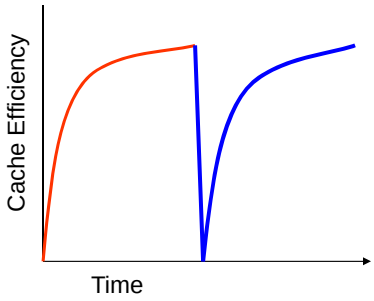




When Blue partition is first dispatched, the data for the Red partition is in L1 cache. Cache efficiency is near 0 to start.

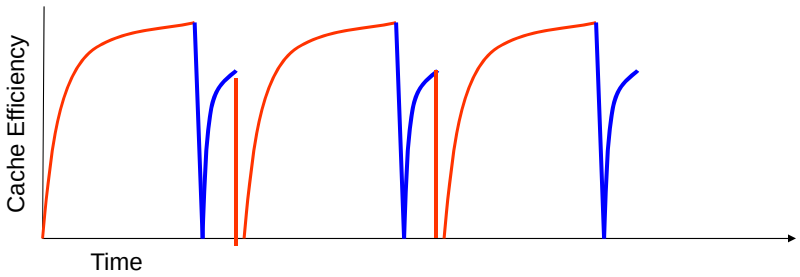
Task Elongation 3





As Blue partition executes, the data for the Blue partition is loaded into L1 cache. Cache efficiency improves… Efficiency of E-unit improves and instruction rate increases.

Impact of Workload



Red partition workload uses full time slice. Blue partition workload is a bursty Workload . It runs for a period << time slice and goes idle. How is the over all processor efficiency effected by the blue partition?

Hiper Dispatch

Problem of cache misses is alleviated by controlling the dispatch of LCPs on specific RCPs.

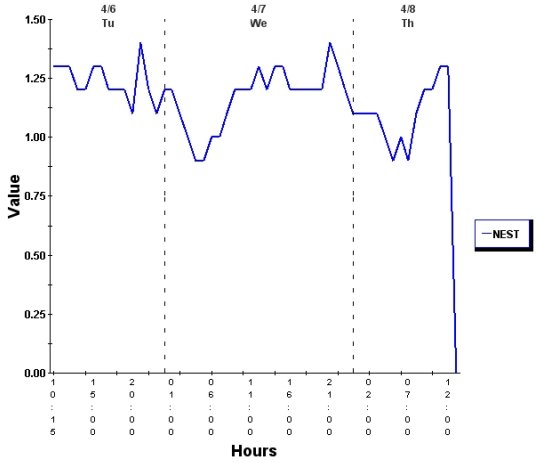
- **Keep LCP-RCP on the same book.**
- **Minimize PRSM dispatching.**
- **Keep LCP on same RCP.**
- **Re-dispatch work units on same processor subset.**

Relative Nest Intensity (RNI)

z10 RNI=(1.0*L2LP + 2.4*L2RP + 7.5*MEMP) / 100

Z196 RNI=1.67*(0.4*L3P + 1.0*L4LP + 2.4*L4RP + 7.5*MEMP) / 100

zEC12 RNI = 2.2*(0.4*L3P + 1.2*L4LP + 2.7*L4RP + 8.2*MEMP) / 100

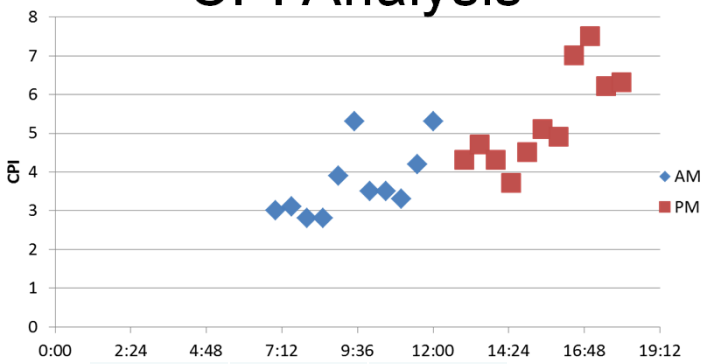


Memory and Workload Characteristics

L1MP	RNI	Workload Hint
<3%	>= 0.75 < 0.75	AVERAGE LOW
3% to 6%	>1.0 0.6 to 1.0 < 0.6	HIGH AVERAGE LOW
>6%	>=0.75 < 0.75	HIGH AVERAGE

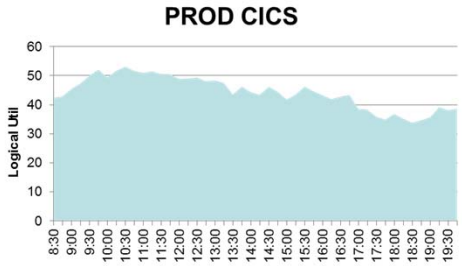
Note that these are initial values and may change.

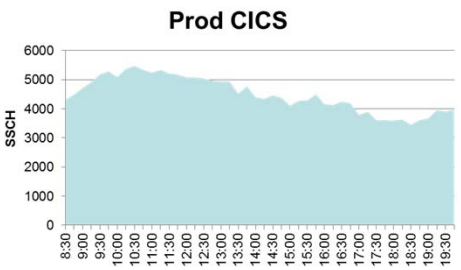
CPI Analysis



	AM	PM
Mean	3.7	5.318182
Variance	0.812	1.529636
Observations	11	11
Pooled Variance	1.170818	
Hypothesized Mean Difference	0	
df	20	
t Stat	-3.50723	
P(T<=t) one-tail	0.001109	
t Critical one-tail	1.724718	
P(T<=t) two-tail	0.002218	
t Critical two-tail	2.085963	

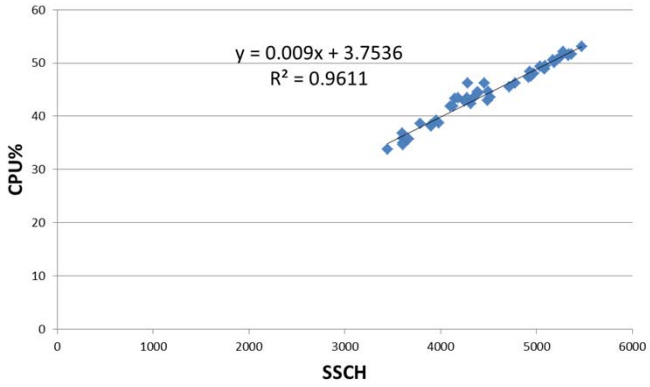
Workload Analysis





Measures of Work
- **Transactions/second**
- **I/Os per second**
  - **EXCPs or SSCH**

Stable Workload



CPU% = F(Work)
= F(SSCH)

Summary

- **Performance is the rate of units of work in processing time.**
- **Architecture is implemented to improve the units of work in time.**
- **There are many contributions of improving units of work in time.**