S. B.

RBC Requirements

&

Specification Review

July 17, 1990

NASA-JPL
C. Buzzell
S. Bellenot
R. Fujimoto
M. Robb

<u>Agenda</u>

⊙ Meeting Objectives

⊙ Specification Outline

⊙ Design Changes From Phase I Prototype

⊙ Specification Review

⊙ Commercial Multiprocessor Hardware Interfaces

⊙ GP1000 Availability

⊙ Time Warp Release Version

<u>Objectives</u>

- ⊙ Review and finalize RBC specification sections 1-4 (functional requirements).

- ⊙ Review section 5 of RBC specification and proposed software interface interface methodology.

- ⊙ Present results of multiprocessor vendor survey and proposed standard hardware interface.

- ⊙ Present status on GP1000 interface task and discuss available options.

- ⊙ Agree on version, and delivery of Time Warp version to be used for modification and incorporation into Phase II program.

Specification Outline

Changes From Phase I Prototype(20)

Spec. ¶	Description	Prototype Module	Current Design
4.3	Total Memory per processor Node	33 Mbytes	129 Mbytes
4.4	RBC Memory allocation	2 non-contiguous Blocks	Segments
4.5	Maximum Number of Processes	1024	256
4.11	Number of Previous States (frames) Retained	31	127
4.14.2	Rollback Method	Background	Foreground + WB read clearing
4.7	Dynamic Process Creation	Not Spec'd	Yes (segments)
4.9	Dynamic Process Destruction	Not Spec'd	Yes (segments)
4.10	Dynamic State Growth/Shrinkage	Not Spec'd	Yes (segments)

Number of Previous States(4.11)

<u>WAS</u>

31 Frames

<u>Phase II</u>

127 frames

- Assume granularity of 10 - 18 msec per event.

- Assume Mark (state save) occurs at the completion of each event.

- Assume GVT updates occur 1 per second (archiving and release of archived frames).

- Assume zero time associated with any processor operation except event processing (i.e. worst case for frame usage).

Then:

RBC needs to maintain sufficient frame reserve to allow the simulation to progress without exhausting all available frames.

$$\text{Number frames required} = \frac{\text{Time between GVT update}}{\text{Mark Frequency (Granularity)}}$$

$$= \frac{1}{10 \text{ msec}} \text{ to } \frac{1}{18 \text{ msec}}$$

$$= 55 \text{ to } 100 \text{ frames}$$

Total Memory per Processor or Node (4.3)

<u>WAS</u>

33 Mbytes

<u>Phase II</u>

129 Mbytes

- Follows directly from change in spec. section 4.11

- Maximum state per Node is 1 Mbyte

- Number of previous frames tracked is 127 instead of 31.

Then:

Total state memory required to be available is:

a) Current mark frame   1 Mbyte Max

b) Previous frames   127 Mbytes Max

     127 history frames * 1 Mbyte/ frame

c) Pre-GVT state memory   1 Mbyte Max
      archiving   _____

      Total RBC Memory per node   129 Mbytes Max

RBC Memory Allocation (4.4)

<u>WAS</u> | <u>Phase II</u>

2 Non-contiguous   Allocation based
state blocks   on segments.

Required:

- Both JPL's and Jade's implementation of Time Warp utilize multiple, non-contiguous memory segments to define (allocate) a state vector. Also both utilize dynamic growth and shrinkage of the state vectors.

- JPL has a programmed limit of 100 such segments and current applications utilize them in the low 10's.

- Jade's implementation allows "any number" of state block allocations and dynamic memory segments.

Prototype Capability:

- Allowed up to 2 non-contiguous state blocks be defined for each process. State blocks could reside anywhere in the 32 bit address space of the node processor.

- RBC hardware performed the necessary address translation to map the state blocks into the WB memory.

- Each process required that its state block boundaries be defined.

         2 words for each block * 2 blocks per process * up to 1024 processes = 4 K words

RBC Memory Allocation (Cont'd)

- Entire 32 bit address space covered on 1 Kbyte increments, 22 bits required for each definition. To perform the address comparison and translation required over 50 IC's to implement.

- Each process directly linked to its defined state blocks and the definitions fixed after initialization.

- Node required to inform the RBC of which process was executing (so the correct values for address translation could be loaded) through the use of the NEW PID command.

- Utilizing the prototype technique will not support the requirements for multiple segment definitions without excessive chip counts.

Phase II Capability:

- RBC memory allocation based on segment definitions which are not directly linked to the processes they serve.

- Processes may consist of multiple segments and the segments need not be contiguous in memory. A single segment cannot contain multiple processes.

- Segments must be at least 1 Kbyte in size and multiples of 1 K bytes.

- Up to 256 segments may be defined.

- Requires that the segment be identified for RBC commands (Mark, Rollback, Advance....) but not for reads and writes.

Maximum Number of Processes (4.5)

<u>WAS</u>   <u>Phase II</u>

1024   2 5 6

-       Follows directly from paragraph 4.4 RBC memory allocation

-       256 segments allowed and each segment can be a separate process. Thus 256 processes per processor are permitted.

-       Change to 256 based on:

a.   Comments at the completion of Phase I indicated that 1024 processes utilized in Phase I was a high estimate and that T.W. would utilize substantially fewer processes.

b.   Printed Circuit board area utilization and form factor: To accommodate existing processors which support VME I/F's the form factor for the RBC module has been switched from 9U X 400mm (14.5 X 15.7 inches) to 6U X 160 mm (10.3 X 7.2 inches). This represents a 65% reduction in board area.

Reducing segment ID width from 10 to 8 ($2^8 = 256$), allows single chip implementation of segment ID storage and configuration storage data path buffers instead of double thus reducing chip count.

c.   Fujimoto has estimated that 64 processes per processor is adequate for most applications.

d.   The need for adding the capability to have multiple segments and dynamic segments is perceived to have higher priority than number of processes

Rollback Method (4.14.2)

<u>WAS</u>

Background

<u>Phase II</u>

Foreground +
Read WB

Prototype:

- Relied on clearing all rolled back frames for <u>ALL</u> WB addresses in the current process.

- Rollbacks could not be queued and the current rollback must finish prior to either a Mark or additional Rollback.

- Implementation Advantages:

a. High speedup when rollbacks are infrequent :

Work load (clearing of the WB) only occurs when the rollback occurs (i.e. infrequently). If rollbacks are infrequent with respect to Marks then speedup versus Phase II method is maximized.

b. High speedup when processor " dead" time is available during the rollback or when long periods elapse after the RB command and Mark or additional RB:

Rollback is a background task and operates when other tasks are not actively using the RBC but are still processing.

c. Conceptually simple

Rollback Method (cont'd)

- Implementation Disadvantages:

    a. Design complexity vs conceptual simplicity

    b. High chip count to implement

    c. If Mark or additional RB commands follow closely, implementation becomes an expensive foreground task.

Rollback Method (cont'd)

Phase II:

- Decrement CMF counter in foreground based on RB distance.

- Delay clearing of the WB memory bits until the addresses are read from the WB during normal foreground operations

- Implementation Advantages:

    a. Rollback used data.   Only addresses which are utilized (i.e. read) are updated. Time is not wasted rolling back unused WB addresses.

    b. Combining of rollback commands.   Addresses are updated when read, rollbacks are combined to give a total rollback distance for addresses which are accessed infrequently.

    c. Sharing of WB accesses. Combining the rollback with the WB read, reduces the time to perform each operation independently.

    d. Requires lower chip count than for background clearing

- Implementation Disadvantages:

    a. Conceptually more difficult than background clearing

    b. Requires insertion of an operation in the read/write path of the processor. This must be carefully designed to prevent a "decision hit" penalty on every read/write cycle.

Specification Review

Commercial Multiprocessor Hardware Interfaces

- Twelve (12) vendors surveyed

BBN Advanced Computers	Intel Scientific Computers	NCUBE
Cogent Research	Kendall Square Research	Paracom Inc.
Concurrent Computer Corp.	Meiko Scientific	Sun Microsystems
Encore Computer	Myrias Computer Corp	Topologix

- Two vendors failed to respond

Topologix

Kendall Square Research

- Very few support a standard bus configuration at the node level.

- Of those that do or are planned VME is the accepted standard.

- Processor Mix

Transputer	68020	80386	Other
Cogent	Myrias	Intel IPSC/2	NCUBE
Topologix	BBN GP1000		Intel IPSC/860
Meiko Scientific	Sun 3		Encore NS32332
Paracom			Concurrent (prop)
			BBN TC2000 (88100)
			Sun4 (prop)

- Interface capabilities

Direct VME link available:	VME link available easily through adapter:	VME link difficult or performance Prob.
BBN TC2000	Intel IPSC/2	Intel IPSC /860
Paracom	BBN GP1000	Myrias
	Cogent	Meiko Scientific
		NCUBE
		Encore
		Concurrent

Some Sun's

{DMA channel
{ long setup.
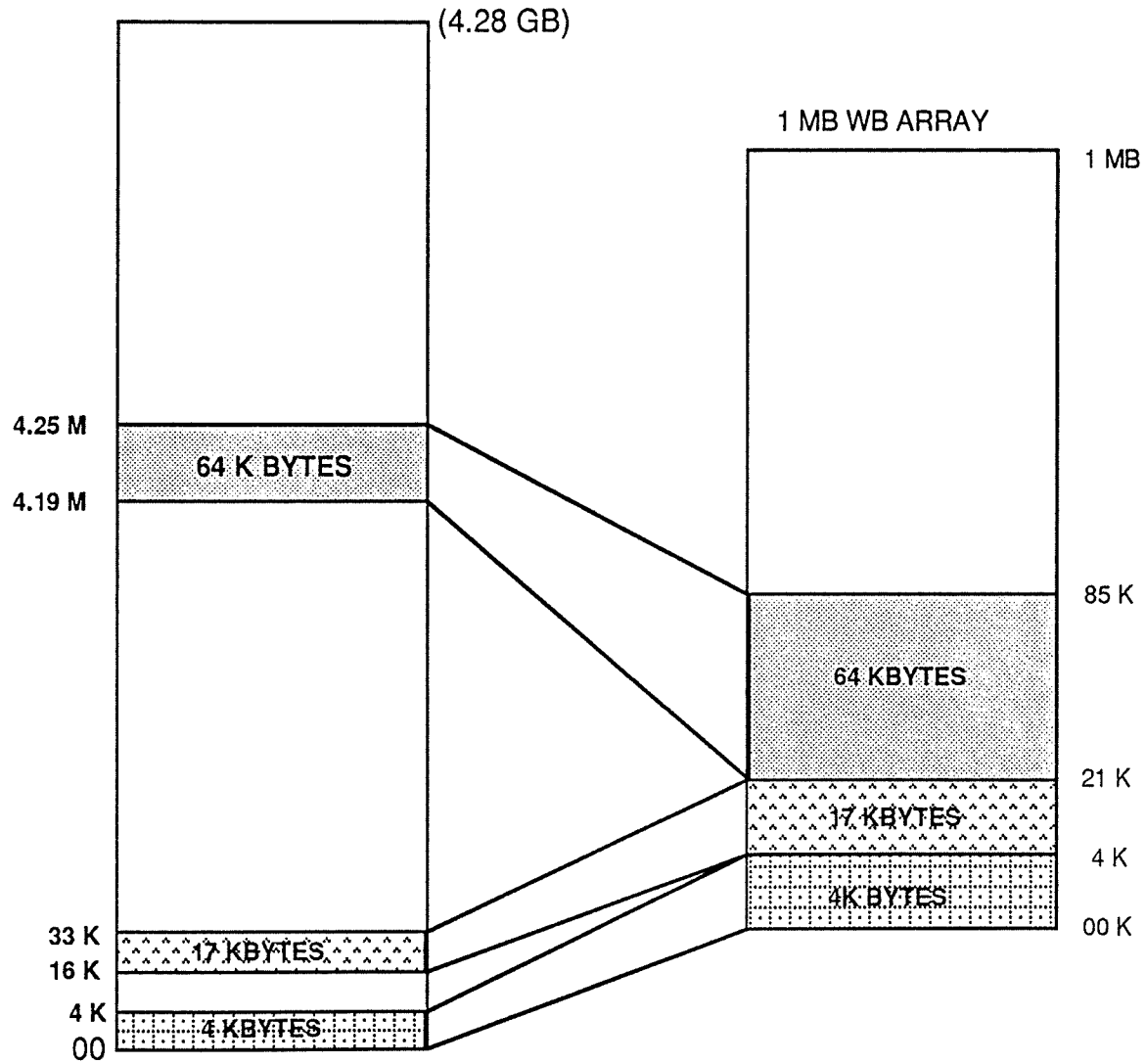
} serial link

shared memory
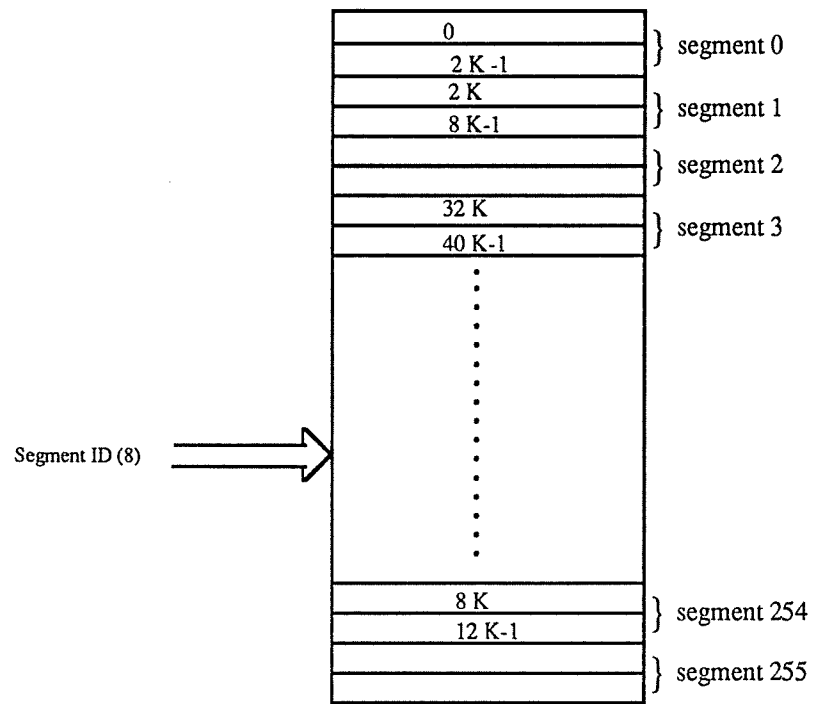
GP1000 Availability and H/W Interface

- GP1000 availability in Mountain View Facility agreed to by BBN is no longer possible due to closure of the field office.

- IPT currently waiting on BBN for designation of GP1000 and integration location to fulfil the agreement.

- IPT has engaged a local consultant to assist in resolution of switch and VME latency concerns on the GP. Consultant was a design engineer on the GP1000 node processor board (R. Quiros).

Time Warp Release Version

- Need to provide to IPT a copy of Time Warp which is to be the version designated as deliverable for the Phase II development. IPT will modify the TW code to interface it with the RBC hardware.

- Source modifications performed by IPT consultants and engineering staff.

- Need source listing and any available documentation

- Need test Time Warp application for the purpose of testing RBC and TW modifications.

- Date for delivery: open

- Media for delivery: open

- Delivery is direct to IPT for configuration control.

68020 'A' & 'B' DEFINITION

(4.28 GB)

1 MB WB ARRAY

1 MB

4.25 M

64 K BYTES

4.19 M

85 K

64 KBYTES

21 K

17 KBYTES

4 K

4 KBYTES

00 K

33 K

17 KBYTES

16 K

4 K

4 KBYTES

00

0	} segment 0
2 K -1	
2 K	} segment 1
8 K-1	
	} segment 2
32 K	} segment 3
40 K-1	

Segment ID (8) ⇒

8 K	} segment 254
12 K-1	
	} segment 255

Segment Definition Table Example,

0

1 K   0   } segment 0

2 K   0

3 K   5   — segment 5

4 K   4

5 K   4   } segment 4

6 K   4

7 K   10   — segment 10

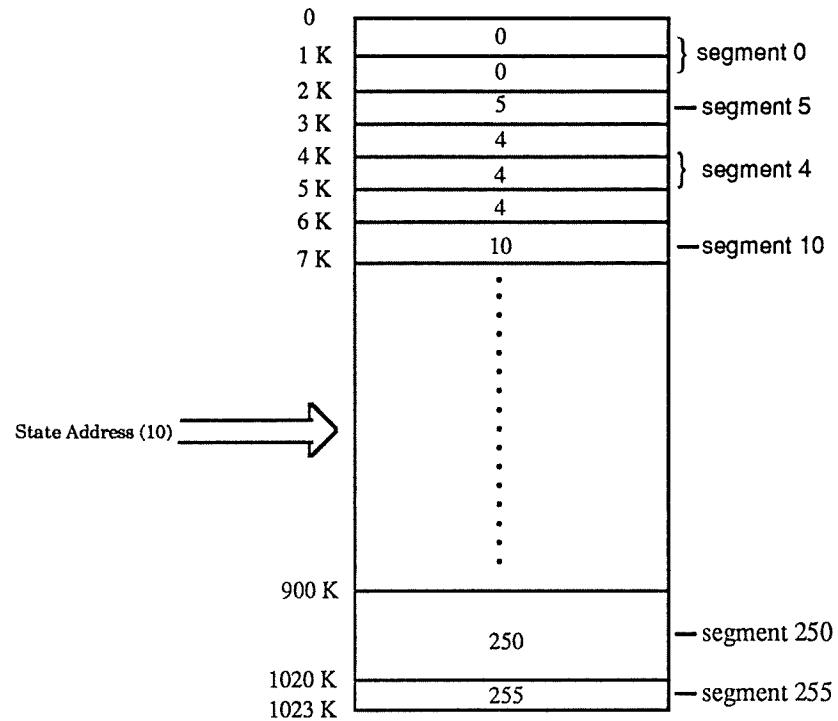State Address (10)

900 K

250   — segment 250

1020 K

255   — segment 255

1023 K

Figure 4, Segment Encoder Table Example