國立中山大學資訊工程學系

93 學年度第 2 學期博士班資格考試　計算機結構

1. A multiprocessor system uses a set of processors with memories attached directly to the processors and interconnected with a bus. Assuming the following characteristics for a machine with 64-byte cache blocks:

Total data miss rate (assume instruction miss rate is negligible)	2%
% misses to private data (used only by this processor)	60%
% misses to shared data that is unowned in a remote cache/memory	20%
% misses to shared data that is dirty in a remote cache	80%

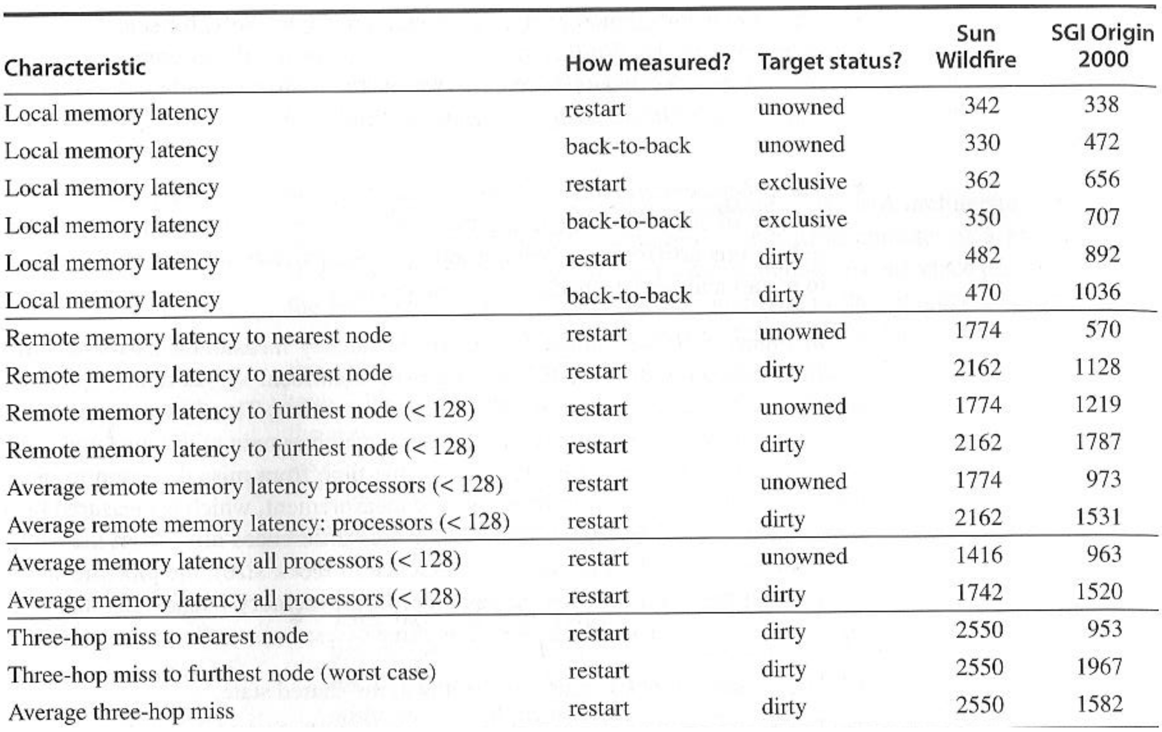
Assume a local memory miss takes 100ns until processor restart. For remote misses that must use the bus, use the E6000 (Wildfire) restart memory miss times from top portion of Fig. 1.

1.1 (5%) Find the average miss time (denoted by T_{avg_miss}) for this design.

1.2 (5%) Let the answer for 1.1 is denoted by T_{avg_miss}. Assume a 1GHz clock rate, a CPI of 1.0 when the cache hit rate is 100%, and that a load or store is issued every other clock cycle. If the processor is stalled during a cache miss until processor restart, what is the effective CPI (denoted by CPI_{eff})? What is the effective rate at which loads or stores are issued (R_{eff_LS})?

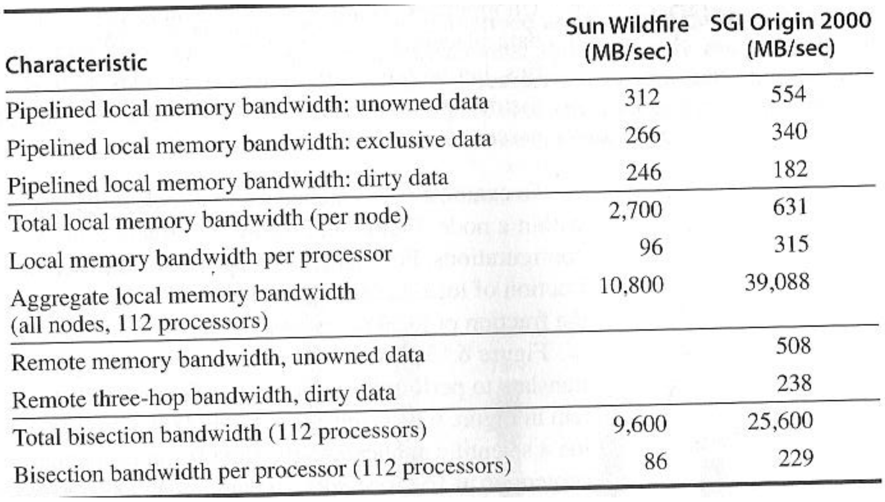
1.3 (5%) Assume a split-transaction bus with a request and acknowledge for all bus transactions. If a bus or acknowledge requires 16 bytes of bus bandwidth and a data transfer requires a total of 80 bytes, what is the bandwidth demand on the shared bus per processor?

1.4 (5%) Using the results from 1.2, and the assumption from 1.3 about bandwidth requirements, determine how many processors could share a bus with the bandwidth characteristics of the E6000(Wildfire) bus as shown in the top portion of Fig. 2. Assume that the processors should not consume more than 80% of the unowned or exclusive data bandwidth.



Characteristic	How measured?	Target status?	Sun Wildfire	SGI Origin 2000
Local memory latency	restart	unowned	342	338
Local memory latency	back-to-back	unowned	330	472
Local memory latency	restart	exclusive	362	656
Local memory latency	back-to-back	exclusive	350	707
Local memory latency	restart	dirty	482	892
Local memory latency	back-to-back	dirty	470	1036
Remote memory latency to nearest node	restart	unowned	1774	570
Remote memory latency to nearest node	restart	dirty	2162	1128
Remote memory latency to furthest node (< 128)	restart	unowned	1774	1219
Remote memory latency to furthest node (< 128)	restart	dirty	2162	1787
Average remote memory latency processors (< 128)	restart	unowned	1774	973
Average remote memory latency: processors (< 128)	restart	dirty	2162	1531
Average memory latency all processors (< 128)	restart	unowned	1416	963
Average memory latency all processors (< 128)	restart	dirty	1742	1520
Three-hop miss to nearest node	restart	dirty	2550	953
Three-hop miss to furthest node (worst case)	restart	dirty	2550	1967
Average three-hop miss	restart	dirty	2550	1582

Fig. 1. Comparison of memory access latencies (in ns) between the Sun Wildfire prototype (using E6000 nodes) and an SGI origin 2000.



Characteristic	Sun Wildfire (MB/sec)	SGI Origin 2000 (MB/sec)
Pipelined local memory bandwidth: unowned data	312	554
Pipelined local memory bandwidth: exclusive data	266	340
Pipelined local memory bandwidth: dirty data	246	182
Total local memory bandwidth (per node)	2,700	631
Local memory bandwidth per processor	96	315
Aggregate local memory bandwidth (all nodes, 112 processors)	10,800	39,088
Remote memory bandwidth, unowned data		508
Remote three-hop bandwidth, dirty data		238
Total bisection bandwidth (112 processors)	9,600	25,600
Bisection bandwidth per processor (112 processors)	86	229

Fig. 2. Comparison of memory bandwidth measurements (in MB/sec) between the Sun Wildfire prototype (using E6000 nodes and an SGI origin 2000.

2. Assume a system around a processor with in-order execution that runs at 1.1 GHz and has a CPI of 0.7 excluding memory accesses. The only instructions that read or write data from memory are loads (20% of all instructions) and stores (5% of all instructions). The memory system for the computer is composed of a split L1 cache that imposes no penalty on hits. Both the I-cache and D-cache are direct mapped and hold 32KB each. The I-cache has a 2% miss rate and 32-byte blocks, and the D-cache is write through with a 5% miss rate and 16-byte blocks. There is a write buffer on the D-cache that eliminates stalls for 95% of all

writes.

The 512 KB write-back, unified L2 cache has 64-byte blocks and an access time of 15 ns. It is connected to the L1 cache by a 128-bit data bus that runs at 266 MHz and can transfer one 128-bit word per bus cycle. Of all memory references sent to the L2 cache in this system, 80% are satisfied without going to main memory. Also, 50% of all blocks replaced are dirty.

The 128-bit-wide main memory has an access latency of 60 ns, after which any number of bus words may be transfer at the rate of one per cycle on the 128-bit-wide 133 MHz main memory bus.

2.1 (5%) What is the average memory access time for instruction accesses

2.2 (5%) What is the average memory access time for data reads? What is the average memory access time for data writes?

2.3 (5%) What is the overall CPI, including memory accesses?

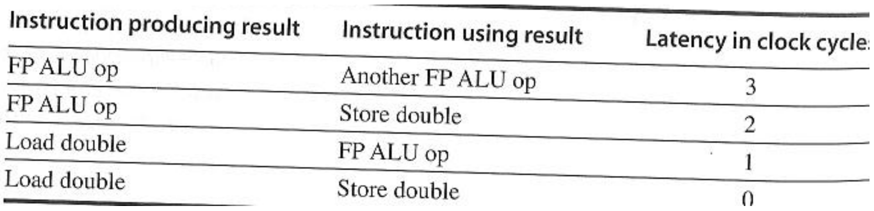
2.4 (5%) You are considering replacing the 1.1 GHz CPU with one that runs at 2.1 GHz, but is otherwise identical. How much faster does the system run with a faster processor? Assume the L1 cache still has no hit penalty, and that the speed of the L2 cache, main memory, and buses remains the same in absolute terms (e.g., the L2 cache still has a 15 ns access time and a 266 MHz bus connecting it to the CPU and L1 cache).

3. The following loop computes $Y[i] = a \times X[i] + Y[i]$, the key operation in many DSP applications. Assume the pipeline latencies are given in Fig. 3 and a 1-cycle delayed branch.

```
loop:       L.D         F0, 0(R1)       ; load X[i]
            MUL.D       F0, F0, F2      ; multiply a*X[i]
            L.D         F4, 0(R2)       ; load Y[i]
            ADD.D       F0, F0, F4      ; add a*X[i]+Y[i]
            S.D         0(R2), F0       ; store Y[i]
            DSUBUI      R1, R1, #8      ; decrement X index
            DSUBUI      R2, R2, #8      ; decrement Y index
            BNEZ        R1, loop        ; loop if not done
```

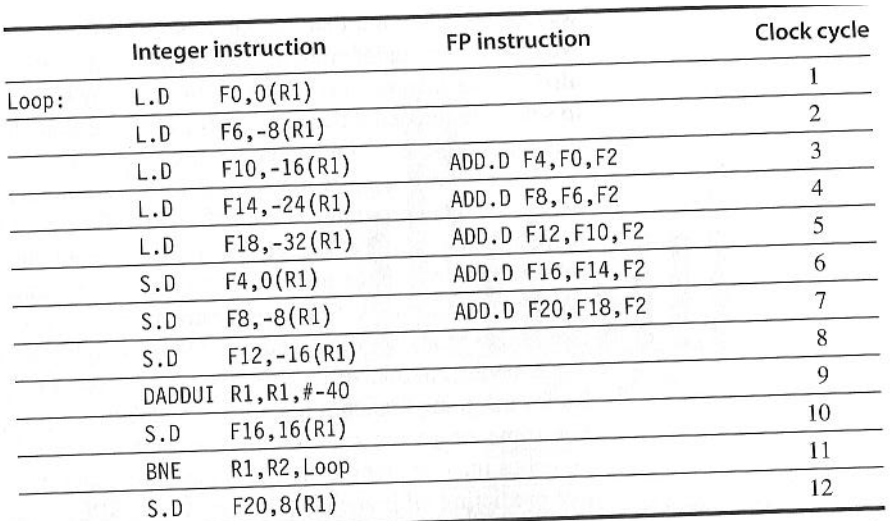
3.1 (10%) Assume a single-issue pipeline. Unroll the loop as many times as necessary to schedule it without any delays, collapsing the loop overhead instructions. Show the schedule. What is the execution time per element? How many instruction issue slots are unused?

3.2 (10%) Assume a dual-issue processor as in Fig. 4. Unroll the loop as many times as necessary to schedule it without any delays, collapsing the loop overhead instructions. Show the schedule. What is the execution time per element? How many instruction slots are unused?



Instruction producing result	Instruction using result	Latency in clock cycle:
FP ALU op	Another FP ALU op	3
FP ALU op	Store double	2
Load double	FP ALU op	1
Load double	Store double	0

Fig. 3. Latencies of FP operations.



	Integer instruction	FP instruction	Clock cycle
Loop:	L.D F0,0(R1)		1
	L.D F6,-8(R1)		2
	L.D F10,-16(R1)	ADD.D F4,F0,F2	3
	L.D F14,-24(R1)	ADD.D F8,F6,F2	4
	L.D F18,-32(R1)	ADD.D F12,F10,F2	5
	S.D F4,0(R1)	ADD.D F16,F14,F2	6
	S.D F8,-8(R1)	ADD.D F20,F18,F2	7
	S.D F12,-16(R1)		8
	DADDUI R1,R1,#-40		9
	S.D F16,16(R1)		10
	BNE R1,R2,Loop		11
	S.D F20,8(R1)		12

Fig. 4. The unrolled and scheduled code as it would look like on a superscalar MIPS for the loop: *for (i=1000; i>0; i=i-1) x[i] = x[i]+s;*

4. Consider a branch-target buffer that has penalties of 0, 2, and 2 clock cycles for correct conditional branch prediction, incorrect prediction, and a buffer miss, respectively. Consider a branch-target buffer design that distinguishes conditional and unconditional branches, storing the target address for a conditional branch and the target instruction for an unconditional branch.

4.1 (5%) What is the penalty in clock cycles when an unconditional branch is found in the buffer?

4.2 (5%) Determine the improvement from branch folding for unconditional branches. Assume a 90% hit rate, an unconditional branch frequency of 5%, and 2-cycle penalty for a buffer miss. How much improvement is gained by this enhancement? How high must the hit rate be for this enhancement to provide a performance gain?

5. An image array of 24-bit picture elements (pixels), each comprised of three 8-bit unsigned integers, representing read, green, and blue color brightness. Larger values are brighter.

5.1 (5%) Brighten the two pixels E5F1D7 and AAC4DE by adding 20 to each color component using unsigned arithmetic and ignoring overflow to maintain ta fixed total instruction-processing time. The values are give in hexadecimal. What are the resulting pixel values? Are the pixels brightened?

5.2 (5%) Repeat 5.1 but use saturation arithmetic instead. What are the resulting pixel values? Are the pixels brightened?

6. Answer the following questions.

6.1 (10%) Explain the three different hazards: control hazard, data hazard, and structure hazards. For each of the three hazards, give a possible solution.

6.2 (10%) Explain the differences of branch prediction buffers and branch target buffers.