CENG3420 Homework 3

Due: Apr. 23, 2017

Q1 (**10%**) Explain how page offset, page number, virtual address and physical address are associated to each other.

Q2 (**15%**) Elaborate advantages and disadvantages of **LARGE** page size.

Q3 (**10%**) Here are two different I/O systems intended for use in transaction proceeding:

- System A can support 15,000 I/O operations per second and use the processor with MIPS rate of 50.
- System B can support 1,000 I/O operations per second and use the processor with MIPS rate of 500.

Assume that each transaction requires 5 I/O operations and each I/O operation requires 10,000 instructions. Ignoring response time, what is the maximum transactions per second for each system.

Q4 (**10%**) For the following code:

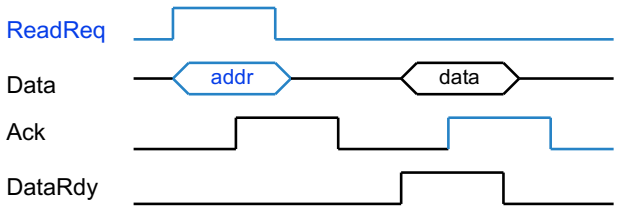
```
for (int i = 0; i < N; ++i) {
        sum[i] = 0;
        for (int j = 0; j < i; ++j) {
                sum[i] = (sum[i] + array[j]) % N;
        }
}
```

Clearly the code takes $\mathcal{O}(N^2)$ time. We would like to improve the actual running time. Which of these strategies would you recommend. Why? (2% for choice and 8% for reason)

```
// Option1
for(int i = 0; i < N; ++i) {
        sum[i] = 0;
        parallel_for(int j = 0; j < i; ++j) {
                sum[i] = (sum[i] + array[j]) % N;
        }
}

// Option2
parallel_for(int i = 0; i < N; ++i) {
        sum[i] = 0;
        for (int j = 0; j < i; ++j) {
                sum[i] = (sum[i] + array[j]) % N;
        }
}
```

Q5 (**18%**) Considering a scenario that data is transferred from memory to I/O devices. Complete the following Asynchronous Bus Handshaking Protocol.



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2.
3.
4.
5.
6.
7. I/O device sees `DataRdy` go low and drops `Ack`

Q6 (**10%**) In the design of a multi-core processor, there are fixed on chip cache resources. We assume maximum of n cores can be designed with those resources. Let k be the real designed core number ($r = \frac{n}{k}$ is integer.) Define a speed up factor $s(r)$ as sequential performance gain by using the resources equivalent to r cores to form a single core, and obviously $s(1) = 1$. Given f the fraction of software that is parallelizable across multiple cores, prove the speed up of the multi-core processor in terms of f, r, n, and $s(r)$ is

$$S(f, r, n) = \frac{1}{\frac{1-f}{s(r)} + \frac{f \times r}{n \times s(r)}} \tag{1}$$

Q7 (**20%**) For the following loop code,

```
lp: lw      $t0, 0($s1)
    lw      $t1, 0($s2)
    addu    $t0, $t0, $t1
    sw      $t0, 0($s1)
    addi    $s1, $s1, -4
    addi    $s2, $s2, -4
    bne     $s1, $0, lp
```

1. (**4%**) Write down the 4 times unrolled code.
2. (**16%**) Schedule the unrolled code and fill the table (**you are free to add more rows**).

Q8 (**7%**) Name 3 cache enhancement techniques and elaborate them.

	ALU or Branch	Data Transfer	cc
			1
			2
			3
			4
			5
			6
			7
			8