

國立臺灣科技大學  
九十學年度碩士班招生考試試題

系所組別：資訊工程系在職專班  
科 目：資訊工程實務

1. Explain the architecture of the following PLDs: (12%)

- (1) PAL(GAL)
- (2) PLA
- (3) FPGA

2. Assume the 2'S complement format and that  $A=00110010$   
 $B=01001010$   $C=11101001$  and Find (12%)

- (a)  $A+B$
- (b)  $C+B$
- (c)  $C-A$

3. Explain the following terms applied to communication patterns in a message-passing network. (12%)

- (a) Unicast versus multicast
- (b) Broadcast versus Conference
- (c) Channel traffic or network traffic

4. Describe the differences between probability and Fuzzy Set. (14%)

5. Please fill in the blank in the followings: (20%) \* 填充題答案請務必寫於答案卷上，並標示題號依序作答。  
Let S be a counting semaphore. To implement it in terms of binary semaphore we need the following data structure:

```
Var S1: binary-semaphore;
    S2: binary-semaphore;
    S3: binary-semaphore;
    C: integer;
```

Initially  $S1=S3=1$ ,  $S2=0$ , and the value of integer C is set to the initial value of the counting semaphore S.

The *wait* operation on the counting semaphore S can be implemented as follows:

```
wait(S3);
  (1) _____;
  C:=C-1;
  if C < 0 then
    begin
      (2) _____;
      (3) _____;
    end
  else (4) _____;
  signal(S3);
```

The *signal* operation on the counting semaphore S can be implemented as follows:

```
wait(S1);
  C:=C+1;
  if C <= 0 then (5) _____;
  signal(S1);
```

The S3 semaphore has no effect on *signal*(S), it merely serializes the *wait*(S) operations.



國立臺灣科技大學  
九十學年度碩士班招生考試試題

系所組別：資訊工程系在職專班  
科 目：資訊工程實務

6. Please list the four necessary conditions for the deadlock.(8%)
7. (a) Does each thread have Program counter and stack pointer? (12%)  
(b) If you use a multi-thread Process to implement a producer/consumer problem, do you need a synchronization mechanism to solve this problem?  
(c) In a multi-thread Process, does the process and threads share the same address space?  
(d) Does thread has own registers?
8. What is the best data structure (Stack, List, Tree) do you used in the following scenarios? (10%)
  - a. Nested hardware interrupt handling
  - b. The implementation of a TOURNAMENT sort
  - c. Data used in a LISP program
  - d. Dynamically created symbol table for fastest search
  - e. Free memory allocation
  - f. Storage allocation for local variables in a Recursive Programming
  - g. Implementing a chaining method in a hash table for collision resolution
  - h. Implement an index file in a database
  - i. Convert an infix expression (i.e.  $a+b$ ) to a prefix expression (i.e.  $+ab$ )
  - j. Handling multiple precision of large decimal numbers operations(add, subtract)



186