1999

p6q2

p13q2

SWM

Computer Design - Qu C. (Part Ib and Diploma)

a) Serial data transmission sends one bit at a time. Parallel data transmission sends several bits simultaneously.

b) ASCII 'A' = 65 in decimal = 01000001 in binary
the start bit is a 1 and the stop bit a 0 so the bits are
sent serially in this order: 1,0,1,0,0,0,0,0,1,0

c) To read a burst of data from DRAM the start address is first presented and the DRAM reads a row of data. The words within this row can then be read out by changing the low order bits of the address (column selection). Since the initial read takes much longer that the column word access, the initial address is held on the bus for several clock cycles whilst the row is read and then the low order bits of the address are changed on a per cycle basis to read all of the words in a quick burst.

d) A cache is a small fast memory used to store values which are likely to be used in the near future. A cache memory is divided into cache lines, each line holding several words from consecutive addresses. A cache line fill, therefore, requires a sequence of words to be read from the DRAM which can be achieved using a DRAM burst read.

Computer Design - Qu D. (Part Ib only)

~~a) register file, first level cache, second level cache, DRAM, swap disk~~

~~b) A register file allows several intermediate results to be stored close to the processor where as an accumulator may only store one result. Consequently an accumulator machine will perform far more memory accesses than a register machine.~~

~~c) register code:~~

```
                mov r0,#0               ; r0 := 0 (r0 holds a)
                mov r1,#1               ; r1 := 1 (r1 holds b)
                mov r2,#0               ; r2 := 0 (r2 holds i)
        loop:   add r0,r1,r0            ; r0 := r0 + r1
                sub r0,r1,r1            ; r1 := r0 - r1
                add r2,#1,r2            ; r2 := r2 + 1
                cmp r2,#5               ; status from r2-5
                blt loop                ; if r2<5 goto loop
```

~~accumulator code:~~

```
                lda #0                  ; clear accumulator
                sta 10                  ; store acc at address 10 (a)
                sta 12                  ; store acc at address 12 (i)
                inc                     ; acc++
                sta 11                  ; store acc at address 11 (b)
        loop:   lda 10                  ; acc:=a
                add 11                  ; acc:=acc+b
                sta 10                  ; a:=acc
                sub 11                  ; acc:=acc-b
                sta 11                  ; b:=acc
                lda 12                  ; acc:=i
                inc                     ; acc++
                sta 12                  ; i:=acc
                cmp #5                  ; status from acc-5
                blt loop                ; if acc<5 goto loop
```