TUTORIAL #6	Polling, Interrupts and DMA	(SC1006/Cx1106)

6.1 Interrupts

1) Consider the system diagram in Figure 1 of case study notes.
   - The processor UART peripheral in its Serial I/O module is configured to receive data with format of 1 start-bit, 7 data-bits, 1 parity-bit and 1 stop-bit.
   - Each time the UART peripheral receives a character, it'll store the data into a buffer.
   - It will then interrupt the CPU to notify CPU that there is data available.
   - The CPU will then execute the Interrupt Service Routine (ISR) to read the character received.
   - Interrupt Latency is the term used to describe the time between interrupt request and entrance to ISR.
   - The minimum interrupt latency for the CPU is 10 µs and it takes 90 µs for the CPU to execute the instructions in the ISR (read the received data from the buffer).
   - Assume that the UART data is transferred back to back i.e. no delays between each UART packets.

   What is the maximum baud rate that can be supported with this UART interface?

2) The specifications of the system in Figure 1 (case study notes) requires the buttons to be asserted 400 times over a 24-hour period. Given that the processor consumes 0.1mA current when it is idling and 50mA when it is in active mode i.e. running code, answer the following.

   90ms idle, 10ms active. Average current = 0.9*0.1mA + 0.1*50mA = 5.09mA

   (a) Supposed polled IO technique is used to sample the button status. Given that the processor poll the buttons every 100ms, each poll requires the processor to be active for 10ms, what is the average current consumed over the 24-hour period?

   (b) If interrupt-driven I/O is used to sample the button status, what is the average current consumed in the same 24-hour period? Given that it takes 20ms to service each interrupt, including interrupt latency and ISR execution.

   (c) Given that a battery with capacity of 2000mAh means that it can supply 2000mA continuously for 1 hour, or 1000mA continuously for 2 hours. How long would a battery of 2000mAh last when used in (a) and (b) above?

b) Total active: 20 x 10^-3 x 50 x 10^-3 x400 = 0.4A

Total time for service: 400 x 20 x 10^-3 = 8s

Total idle A: (86400-8) x 0.1 x 10^-3 = 8.6392A

Total A = 0.4+8.6392A = 9.0392A

c) a) Average current consumed 5.09mA. Battery life = 2000mAh/5.09mA = 393h =16.4 days

c)b) Average current consumed 0.146mA. Battery life = 2000mAh/0.1046mA = 19120h = 797 days = 2.2years

burst cycle stealing or transparent

1. Type of DMAC transfer mode: simplex, half duplex or full duplex. Determines number of directions for data transfer

6.2 Direct Memory Access (DMA)

2. Transfer rate of slowest module
3. CPU's System bus utilization rate. The higher it is, the less chance for DMAC to use the system bus
4. Serial or Parallel data transfer used. Parallel is faster.

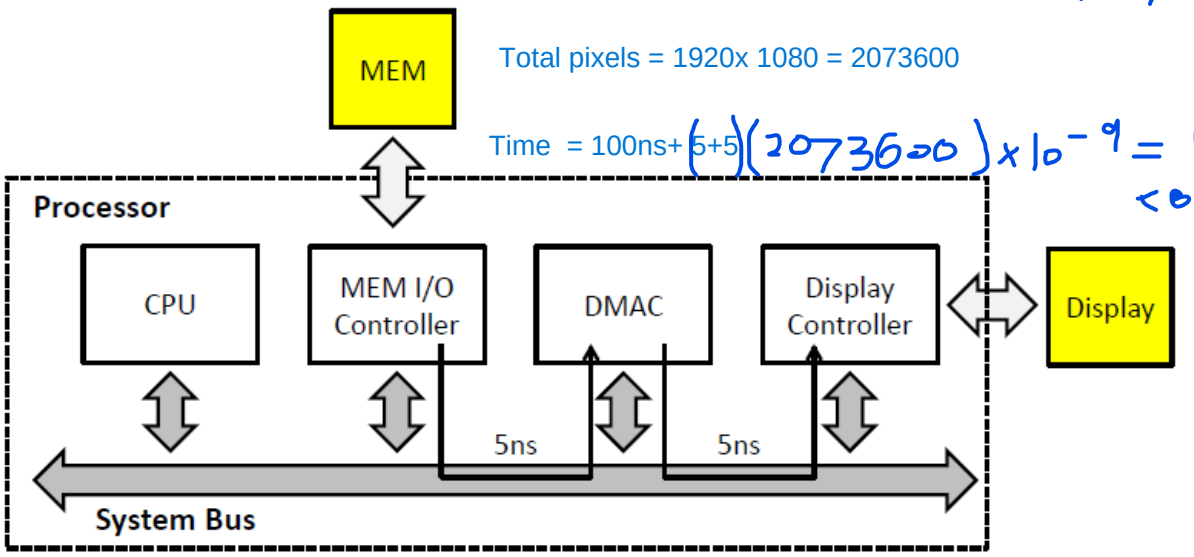
Priority of DMAC wrt CPU. fixed or configurable

3) Given that DMAC and CPU share one system bus, list four possible factors that might affect the transfer rate of a DMAC and explain how they affect the DMAC transfer rate.

4) Consider the system in Figure 1 (Case study notes). Given that

- The processor's system bus is capable of supporting simultaneous transfer of up to 3 bytes of data at one time.
- DMA is used to transfer video data from Memory to Display Controller module.
- Each video pixel data consist of three bytes (Red, Green, Blue) and are transferred simultaneously on the system bus on each bus cycle.
- Each transfer on the system bus takes 5ns and transfer of the bus control between the CPU and the DMAC takes 100ns.
- Assume that DMAC is using Fetch-and-Deposit DMA.
- Note that 1Kbyte = 1024 Byte.

3B/cycle/pixel



Total pixels = 1920x 1080 = 2073600

Time = 100ns+ (5+5)(2073600) x 10^-9 = 0.020736s < 0.0333s ∴ Yes

1 frame = 0.0333333s

(a) Given that the video is output at a rate of 30 frames per second and each video frame has a resolution of 1920x1080 pixels. If burst-mode was used by the DMAC to burst one frame of video data at a time, would the DMAC be able to transfer each video frame completely?

(b) Repeat the calculation if the DMAC is using cycle-stealing mode, assume that DMAC needs to wait at least 5 instructions before it could request for control of the system bus again.

Time = (110 +5(5))(2073600)(10^-9) = 0.279936s>0.03333s
Thus, no

0.461376s

+ bus release + 5* CPU clock cycles wait =
110+ 100ns + 5 x (1/400MHz) = 222.5ns