**Esame di Computer System Design /Calcolatori Elettronici 2 – prof. Mazzocca**

**Prova del 17 maggio 2022**

Un sistema è composto da 3 unità, A, B e C, tra loro collegate mediante due periferiche parallele che interconnettono A con B e A con C rispettivamente.

Il sistema opera in due fasi successive come descritto di seguito:

* Fase 1) A riceve un messaggio di N caratteri da B e poi un messaggio di N caratteri da C (**l’ordine è prefissato, non ci sono sovrapposizioni fra B e C nella prima fase**); dopo averli ricevuti, A verifica se i messaggi ricevuti da B e C sono uguali invocando una subroutine CHECK (LA SUBROUTINE NON DEVE ESSERE IMPLEMENTATA MA SOLO INVOCATA PASSANDO OPPORTUNAMENTE I PARAMETRI SULLO STACK);
* Fase 2) Al termine della fase1, se i messaggi sono uguali, A riceverà T ulteriori messaggi (T fissato a scelta dello studente), ricevendo prima un intero messaggio da B e poi un intero messaggio da C in maniera alternata. Se invece i messaggi ricevuti nella fase 1 sono diversi, A riceverà T ulteriori messaggi, ricevendo prima un messaggio da C e poi da B, in maniera alternata.

**IPOTESI FACOLTATIVA**: nella fase 1 è possibile rimuovere l’ipotesi che l’ordine di arrivo dei messaggi sia prefissato, e gestire l’arrivo di caratteri anche intervallati da B e da C. Si discuta **almeno qualitativamente** questa ipotesi (come cambierebbe il flusso di controllo). **Lo studente può anche scegliere di implementare il sistema A rispetto alla sola ipotesi facoltativa.**

Si progetti e implementi **l’unità A** specificando:

1. *Architettura complessiva*: rappresentazione grafica schematica dell’architettura complessiva del sistema, in termini dei componenti di ciascuna unità (CPU, memoria, bus, dispositivi) e delle relative interconnessioni, in cui siano evidenziati i principali collegamenti e le linee di interruzione previste.
2. *Protocolli*: diagrammi temporali che rappresentino i principali protocolli di comunicazione utilizzati fra i dispositivi (ad es. i protocolli utilizzati per la scrittura e/o la lettura su/da periferica parallela).
3. *Mappa della memoria*: rappresentazione grafica schematica del contenuto della memoria RAM e ROM con riferimento alle aree dati e codice del programma implementato e al vettore delle eccezioni (solo per la specifica unità richiesta).
4. *Descrizione di alto livello del programma implementato*: descrizione, mediante diagramma a blocchi o pseudocodice, dei principali passi effettuati in ciascuno dei moduli software che compongono il programma (si richiede cioè un diagramma separato per il “main” e per ciascuna ISR prevista). In tale descrizione, lo studente deve specificare chiaramente le assunzioni fatte circa il comportamento delle periferiche, ad esempio legato alla gestione di possibili “conflitti” sull’accesso a dati globali e/o alla gestione di possibili “sovrapposizioni” di messaggi dovute alla diversa velocità di elaborazione dei dispositivi coinvolti.
5. *Implementazione*: codice Assembly Motorola 68000 per il sistema progettato. Gli studenti sono invitati a inserire commenti nel codice almeno nelle parti salienti (ad esempio, nella configurazione delle periferiche e nell’utilizzo di variabili globali) per favorire una migliore leggibilità e comprensione dell’elaborato.

Dopo aver sviluppato l’intero progetto, si illustri come cambierebbero l’architettura complessiva e la logica del driver se venisse inserito un componente DMA per la comunicazione fra A e i nodi B e C.

Nota: non è richiesta l’implementazione completa di un nuovo programma, ma lo studente dovrà indicare schematicamente le principali modifiche necessarie al codice assembly già prodotto (è preferibile a tale scopo indicare a parte gli stralci di codice da inserire ove necessario).