

Esercizio 1, Assembler

Preparazione del computer

1. Accedere al computer usando le credenziali dell'aula. Sia per esempio "SI3".
Il nome utente sarà allora **SI3** e la password **unipisaSI3**.
2. Accedere ad internet. Aprire un browser e digitare un url a caso, ad esempio www.google.com
Se il sito si carica correttamente, bene. Passate oltre.
Altrimenti, dovrebbe aprirsi un form dove inserire le proprie credenziali di ateneo. Inserirle correttamente e verificare che ci sia adesso connessione ad internet.
3. Aprite un modulo Google, ad esempio uno di quelli della pagina del corso: <http://tinyurl.com/234tf7hd>
4. Se richiesto, inserite le vostre credenziali di ateneo.
5. Assicurarsi che l'indirizzo email mostrato in alto nel form sia il vostro. Se non lo è, cambiare il login ai servizi di Google (è probabile che l'ultima persona che ha usato il calcolatore non abbia fatto logout).

Test ambiente assembler

Le istruzioni qui contenute servono a verificare che l'ambiente assembler sia pronto all'uso per l'esame.

Il foglio del compito non va girato finché non viene dato il via alla prova d'esame.

1. Aprire la cartella `C:\reti_logiche`
2. Fare doppio click sul file `assembler.code-workspace`
3. Dovrebbe aprirsi una finestra VS Code
 - a. Se appare in basso a destra un avviso che menziona WSL con un bottone "Install and Reload", premerlo.
4. Verificare che in basso a sinistra appaia scritto "WSL"
5. Verificare che l'ambiente abbia i file necessari:
 - a. `assemble.ps1`, `debug.ps1`
 - b. cartella `files/`
 - c. `test-ambiente.s`
6. Aprire un terminale dentro VS Code. Assicurarsi che sia Powershell (`pwsh`).
Altrimenti, usare il + in alto a sinistra del pannello del terminale.
7. Lanciare il comando `./assemble.ps1 test-ambiente.s`
8. Lanciare l'eseguibile `./test-ambiente`
9. Se stampa "Ok.", l'ambiente funziona correttamente.

Download di file ausiliari

I testi d'esame includono link a file ausiliari (esempi, testbench, etc.). Questo link porta ad un file `.zip` su Google Drive, che richiede credenziali di ateneo.

Per vederne il contenuto, il file `.zip` va prima scaricato. Il bottone è in alto a destra.

Uso di VS Code

1. Usare `./assemble.ps1 sorgente.s` per assemblare il sorgente, `./debug.ps1 eseguibile` per debuggare l'eseguibile.
2. Ricordarsi di salvare il file prima di assemblare; un pallino in alto a fianco del nome del file significa che non avete salvato le ultime modifiche.
3. L'estensione per evidenziare il codice assembler è "GNU Assembler Language Support".

Consegnare

1. Ad un certo punto, verrà scritto alla lavagna un codice di 8 caratteri, siano per esempio `abcdefgh`
Allora, all'url tinyurl.com/abcdefgh troverete la form per consegnare il primo esercizio
2. Assicurarsi che l'indirizzo mostrato in alto sia il vostro.
3. Sia per esempio la vostra matricola 12345, allora nominare il file sorgente `12345.s`
4. Troverete il file in `C:\reti_logiche\assembler\`
5. Il form chiederà anche di scegliere una password per l'accesso all'autocorrezione.
Le vediamo in chiaro, quindi non usare password usate altrove, non le vogliamo conoscere.
6. È possibile consegnare più di una volta, apportando modifiche al sorgente.

Commenti

È possibile usare commenti per documentare il codice e chiarire le vostre intenzioni.

Ne terremo poi conto, sia in positivo che in negativo.

Esercizio 2, Verilog

Test ambiente Verilog

Le istruzioni qui contenute servono a verificare che l'ambiente Verilog sia pronto all'uso per l'esame.

Il foglio del compito non va girato finché non viene dato il via alla prova d'esame.

1. Aprire la cartella `C:\reti_logiche`
2. Fare doppio click sul file `verilog.code-workspace`
3. Dovrebbe aprirsi una finestra VS Code
4. Verificare che l'ambiente abbia il file `test-ambiente.v`
5. Aprire un terminale dentro VS Code.
6. Lanciare il comando `iverilog -o test test-ambiente.v`
7. Lanciare il comando `vvp ./test`
8. Se stampa "Ok.", l'ambiente funziona correttamente.

Uso di VS Code

1. Usare `iverilog -o sim [lista di file .v]` per compilare la simulazione, `vvp ./sim` per eseguire la simulazione.
2. Per vedere la waveform dopo una simulazione, lanciare `gtkwave waveform.vcd`
 - a. Si può lanciare il comando in nuovo terminale, per non dover chiudere `gtkwave` ogni volta.
Usare il pulsante refresh in alto per caricare la nuova waveform dopo una nuova simulazione.
3. L'estensione per evidenziare il codice Verilog è "Verilog-HDL/SystemVerilog/Bluespec SystemVerilog".
4. La tilde (~) si fa con `shift+' (a sinistra dell'1)` su tastiera US-International, o `ALT+126` su tastiera italiana.

Consegnare

1. Ad un certo punto, verrà scritto alla lavagna un codice di 8 caratteri, siano per esempio `abcdefgh`
Allora, all'url tinyurl.com/abcdefgh troverete la form per consegnare il secondo esercizio
2. Assicurarsi che l'indirizzo mostrato in alto sia il vostro.
3. Sia per esempio la vostra matricola 12345, allora nominare il file della descrizione `12345d.v` e il file della sintesi `12345s.v`
 - a. Entrambi i file devono contenere tutto il necessario per compilare correttamente insieme ai file forniti.
Verranno compilati e testati separatamente.
4. Troverete i file in `C:\reti_logiche\verilog\`
5. È possibile consegnare più di una volta, apportando modifiche al sorgente.

Commenti

È possibile usare commenti per documentare il codice e chiarire le vostre intenzioni.

Ne terremo poi conto, sia in positivo che in negativo.

Autocorrezione

Dopo la prova, vi sarà comunicato via email quando potrete accedere al sistema di autocorrezione.

Il vostro scopo sarà apportare tutte e sole le minime (dal punto di vista logico) modifiche necessarie a far sì che la vostra prova sia corretta e passi tutti i test. In caso di differenze con il test minime o di nessun conto, correggere lo stesso.

Saranno valutati solo prove che passino tutti i test.

Quando una prova passa i test, il sistema non ne permette più la modifica.

Se ci sono comunque cose da correggere, mandare le modifiche via email.