

I - Các bài tập liên quan đến chu kỳ và thời gian thực thi lệnh

1. Có pipeline

- Chu kỳ xung clock (T) = Thời gian công đoạn dài nhất
- Thời gian thực thi 1 lệnh = 5 (công đoạn) * chu kỳ xung clock
- Thời gian thực thi n lệnh = $5T + (n-1)T$

2. Không pipeline đơn chu kỳ

- Chu kỳ xung clock (T') = Tổng thời gian các công đoạn của lệnh dài nhất (lệnh lw)
- Thời gian thực thi 1 lệnh = chu kỳ xung clock
- Thời gian thực thi n lệnh = nT'

3. Không pipeline đa chu kỳ

- Chu kỳ xung clock (T'') = Thời gian công đoạn dài nhất (giống pipeline)
- Thời gian thực thi 1 lệnh = số công đoạn * chu kỳ xung clock
- Thời gian thực thi n lệnh = $\sum(\% \text{ lệnh} * \text{số công đoạn của lệnh})nT''$

II-Giải quyết xung đột dữ liệu

1. Không sử dụng forwarding

- Nếu lệnh phía sau cần sử dụng thanh ghi, mà thanh ghi đó là kết quả của lệnh phía trước. Thì cần phải thêm lệnh "nop" ở giữa 2 lệnh sao cho lệnh sau cách lệnh trước 2 câu lệnh. Mục đích là để công đoạn ID của lệnh sau bằng với công đoạn WB của lệnh trước.

2. Chỉ sử dụng ALU-ALU forwarding (không có MEM-ALU forwarding)

- Nếu lệnh phía sau cần sử dụng thanh ghi, mà thanh ghi đó là kết quả của lệnh phía trước.
 - + Nếu lệnh trước là lệnh load word (lw): Thì cần phải thêm lệnh "nop" ở giữa 2 lệnh sao cho lệnh sau cách lệnh trước 2 câu lệnh. Mục đích là để công đoạn ID của lệnh sau bằng với công đoạn WB của lệnh trước
 - + Nếu lệnh trước là các lệnh khác: Thì không cần thêm lệnh "nop"
- 3. Sử dụng full forwarding (có cả ALU-ALU và MEM-ALU forwarding)
 - Nếu lệnh phía sau cần sử dụng thanh ghi, mà thanh ghi đó là kết quả của lệnh phía trước.
 - + Nếu lệnh trước là lệnh load word (lw): Thì cần phải thêm lệnh "nop" ở giữa 2 lệnh sao cho lệnh sau cách lệnh trước 1 câu lệnh. Mục đích là để công đoạn ID của lệnh sau bằng với công đoạn MEM của lệnh trước
 - + Nếu lệnh trước là các lệnh khác: Thì không cần thêm lệnh "nop"