**2023년도 1학기**

**논리회로 설계 프로젝트**

**C045-2, C046-3**

* **설계 프로젝트 내용**
* **유전자 네트워크(Gene Network)의 fixed points 찾기**
* **Input:** x1(0), x2(0), …, x8(0)
* **Output:** x1(t), x2(t), …, x8(t)

x(t)는 시간 t에서 논리변수 x의 값을 나타내고, x(0)은 초기값이다. 시간 t에서 8개 유전자들(genes)의 상태는 논리변수 x1(t), x2(t), …, x8(t)로 나타낼 수 있고, 이 유전자들의 t+1에서 값은 시간 t에서 다른 유전자들의 상태에 따라 다음과 같이 결정된다.

**x1(t+1) = x3(t)’∙x7(t)∙x8(t)’**

**x2(t+1) = (x5(t)+x6(t))∙x8(t)’**

**x3(t+1) = x8(t)**

**x4(t+1) = x2(t)∙x7(t)’**

**x5(t+1) = x2(t)+x4(t)**

**x6(t+1) = x3(t)∙x8(t)’**

**x7(t+1) = x2(t)∙x8(t)’**

**x8(t+1) = (x1(t)+x2(t))’∙(x4(t)+x7(t))**

가령, x1 유전자의 t+1에서 값 x1(t+1)은 x3 유전자의 t에서 값 x3(t)의 NOT과 x7 유전자 x7(t)과 x8 유전자의 값 x8(t)의 NOT 값을 AND 논리로 결정한다.

**x1(t+1)=x1(t), x2(t+1)=x2(t), ..., x8(t+1)=x8(t)이 성립할 때 (x1(t), …, x8(t))를 fixed point라고 한다.**

**이 유전자 네트워크를 Verilog로 설계하는데 회로의 input을 초기값으로 설정하고, 시간 t=0,1,…에서 8개 유전자의 상태를 출력으로 하는 회로를 설계하시오. Input을, 그러니까 초기값을 바꾸어 가며 네트워크에 존재하는 fixed points를 모두 찾으시오.**

**Verilog HDL 설계 내용**

1. **유전자 네트워크 설계**
2. 입력한 초기값에 대해 t=0,1,…에서 유전자들의 상태를 출력하는 회로 “gene\_net.v” 설계
3. Fixed point 검사 모듈 설계
4. 출력이 fixed point 인지 확인하는 모듈 fixed\_point\_checker.v 설계

**# 주의: 현재 입력값은 1개이고 이전입력을 기억해서 기능 달성(ex. flip-flop을 통해 구현)**

1. Cycle 검사 모듈 설계

1)2개 이상의 상태가 주기적으로 반복되는 것을 cycle이라 하며, 이 때 cycle을 검사하는 모듈을 설계

**# 주의: 현재 입력값은 1개, 모듈 내에서 과거 입력을 기억해야함**

1. **A,B,C 각각에 대해 test bench 설계**
2. A, B, C에서 설계한 3개 모듈들에 대해 test bench 구현하여 module 당 5가지 이상의 test case를 작성하여 설계한 sub-Module들이 정상 동작하는지 확인
3. 최종 test bench 설계
4. “tb\_top.v” 에 Verilog 강의노트를 참고하여, **A,B,C Module을 Instantiation**한 후 활용하여 그 초기값을 gene\_net 에 입력하고 해당 초기값에 대한 결과를 바탕으로 fixed point 및 cycle 여부를 확인하는 test\_bench 설계

* **A,B,C 모듈에 최종 결과 분석의 원활함을 위해 reset 이나 done 등의 추가 입력을 추가하는 것은 인정**
* **최종 결과는 앞서 구현한 sub\_module 을 Instantiation 한 후 모두 활용하여 설계하여야 인정**
* **모든 코드에 주석을 이용한 간략한 설명 명시 (Code 설명 등)**
  + ModelSim은 한글 지원이 불가해 영문으로 작성, 또는 보고서에서 따로 주석 가능
* **모든 Verilog HDL 파일 중 하나라도 컴파일 에러가 발생하면 실격 처리**
* **보고서 내용**

1. **공학 인증 표지**
2. **문제 설명 (1 page 내로 간략히)**
3. **자신의 시스템 전체 Schematic 그림**

**각 submodule 의 입출력 과 연결 관계**

1. **A,B,C 에서 설계한 Code 분석 (설계 원리, 구조, 동작 따위 자세히)**
2. **모든 sub-module(A,B,C) 들의 Simulation 결과 캡쳐와 분석**
3. **설계한 최종 Test Bench Code 분석, Simulation 결과 캡쳐와 분석**
4. **추가 구현 결과 및 분석**

**Ex) Fixed point, cycle 을 제외한 initial value 에 따른 출력 특징을 발견하여 분석**

**Ex) t가 특정 임계값을 넘기면 유전자네트워크 초기값을 바꿔주는 모듈 구현**

1. **기능별 달성도 평가 (10점 만점 기준)**

**# 중요! 어떤 모듈을 덜 설계 했고, 어떤 모듈을 완벽하게 설계했고, 최종 결과 코드가 실행이 가능한지 가능하지 않은지, 등등 자신의 수행 진행도를 구체적으로 평가.**

* **보고서에 코드 첨부 시, 가시성을 위해 하이라이팅하여 첨부**
  + hilite.me 접속 🡪 ‘Source Code’ 란에 코드 복사/붙여넣기 🡪 ‘Language’ Verilog로 설정 🡪 ‘Highlight!’ 클릭 🡪 하단 ‘Preview’ 란의 하이라이팅된 코드를 복사하여 첨부 (다른 사이트/툴 이용 가능)
* **표지 빼고 10쪽을 넘지 말 것**
* **시뮬레이션 분석을 용이하게 하기 위해 Wave의 Radix 변경 가능**
* **제출 시 유의 사항**
* **제출물 : 보고서 (한글 파일 또는 PDF 파일), 모든 Verilog HDL Codes (.v 파일)**

**하나의 압축 파일로 압축**

**파일 이름 : “학번\_이름\_반.zip”** Ex) “202124109\_이용제\_오전반.zip”

* **제출 기한 : 5월 31일 수요일 자정 (오후 11시 59분)**
  + **시간이 다 되어 제출하면 인터넷 트래픽이 몰려 업로드 오류가 생길 수 있으니 가능한 일찍 제출하기 바람. 정한 시간 뒤에 제출하면 “무조건” delay 감점**
* **제출 방식 : Bb 과제 제출**
* **불이익을 당하지 않게, 제출 전후로 모든 파일 검토 및 확인**