**논리 회로 설계**

**도전 과제**

|  |  |
| --- | --- |
| 프로젝트 명 | Row Dominance, Column Dominance, Petrick Method 구현 |
| 팀 명 |  |
| 문서 제목 | 결과 보고서 |

|  |  |
| --- | --- |
| **Version** | 1.0 |
| **Date** | 2022-05-26 |

|  |  |
| --- | --- |
| **팀원** |  |
|  |
|  |
|  |
|  |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는규 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 논리회로설계 수강 학생 중 도전과제를 수행하는 팀 “이혁규”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “이혁”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 논리회로설계 \_결과보고서.doc |
| **원안작성자** |  |
| **수정작업자** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2022-05-26 |  | 1.0 | 최초 작성 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 개발 내용 및 결과물

### 개발 내용

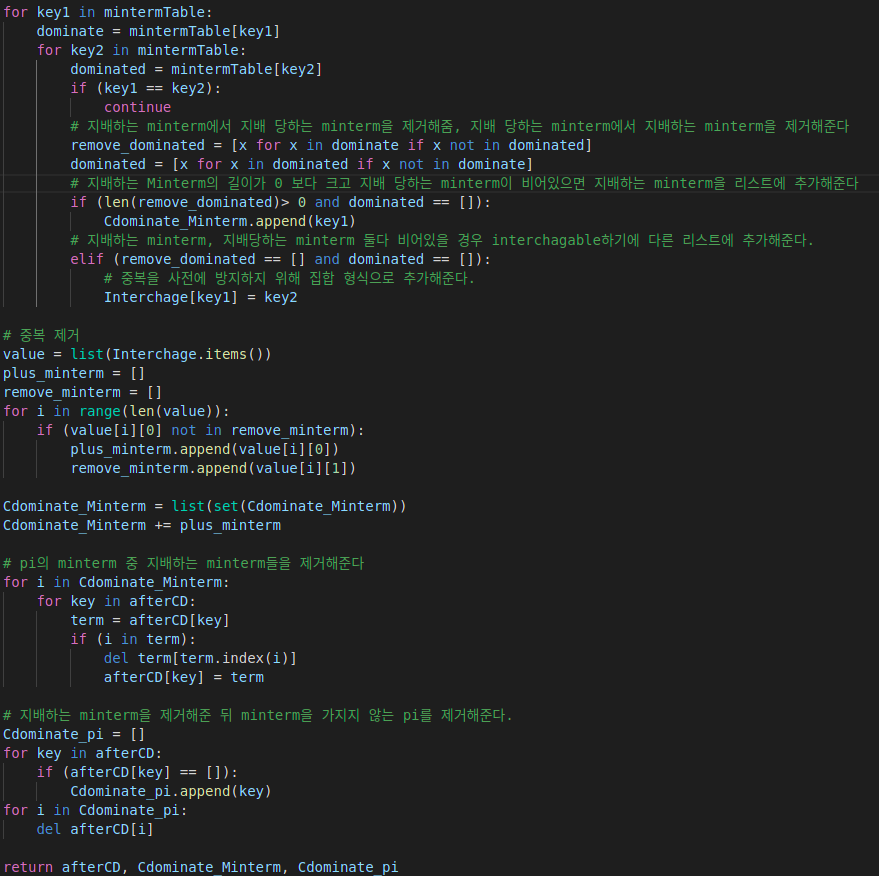
|  |
| --- |
|  |

**- remove\_EPI\_Minterm 함수**

**1. Column Dominance를 하기 전 , EPI가 가지고 있는 minterm들이 NEPI가 가지고 있는 Minterm에 속해있다면 제거해준다.**

**2. 제거 후 NEPI가 가지고 있는 값이 빈 리스트인 경우는 EPI들에게 지배받는 경우(Row Dominance) 이므로 먼저 제거해준다.**

**3. EPI의 Minterm이 제거 된 pi 테이블을 리턴한다.**

**- Column Dominance 함수**

**1. minterm이 row가 되고 pi가 column이 되는 table을 생성한다.**

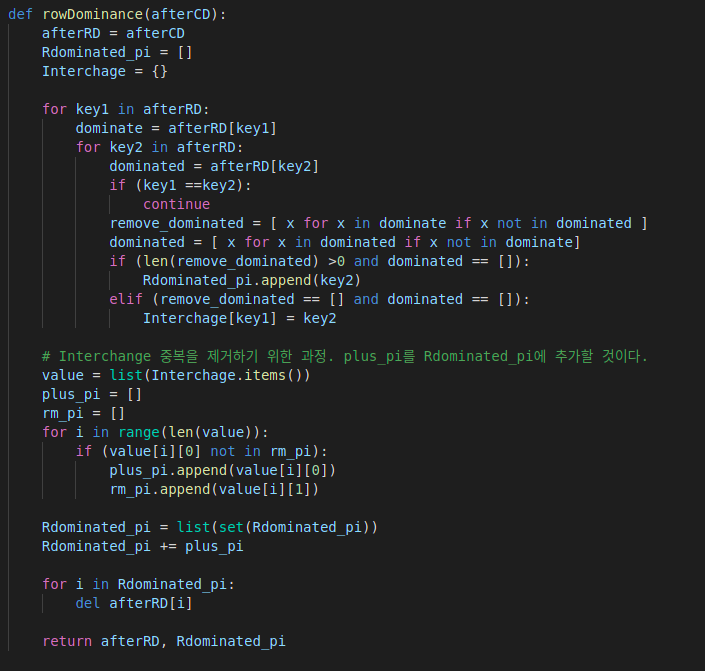
**2. table을 활용해서 지배하는 minterm을 찾아낸다.**

**3. 서로 지배하는 관계(Interchagable)도 추가로 찾아낸 후, 첫번째 나오는 값을 지배하는 minterm으로 설정한다.**

**4. 기존 pi 테이블에서 지배하는 minterm을 제거한다.**

**5. 지배하는 minterm을 제거해준 뒤 minterm을 가지지 않은 pi를 테이블에서 삭제해준다.**

**6. pi table, 지배하는 minterm, 지배하는 pi를 리턴한다.**



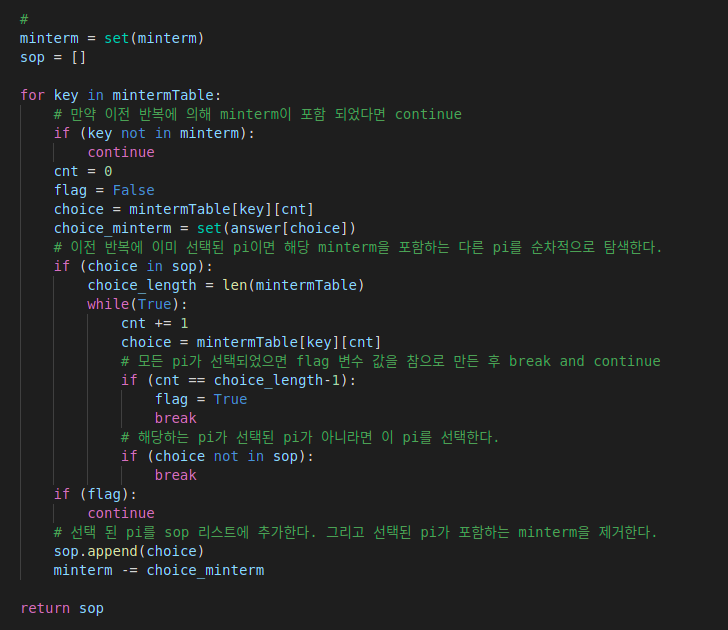
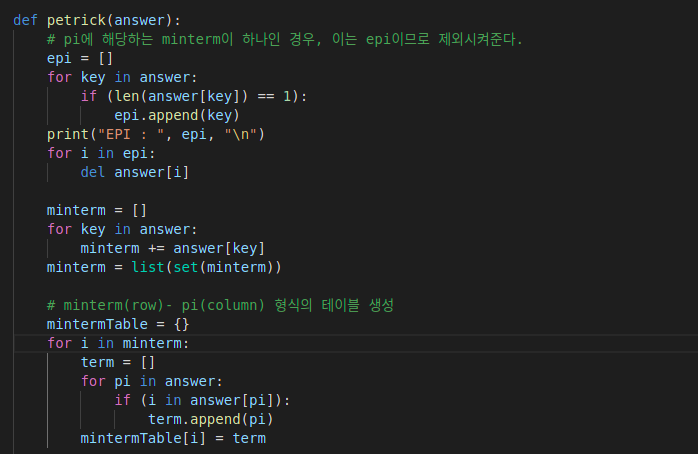
**- Row Dominance 함수**

**1. table을 활용해서 지배당하는 PI을 찾아낸다.**

**2. 서로 지배하는 관계(Interchagable)도 추가로 찾아낸 후, 첫번째 나오는 값을 지배당하는 PI로 설정한다.**

**3. 지배 당하는 PI를 PI 테이블에서 제거한다.**

**4. PI 테이블과 지배당하는 PI를 리턴한다.**

****

**- Petrick함수**

**1. pi에 해당하는 minterm 이 하나인 경우, 다음 findEPI에서 검출되어야 할 EPI이므로 Petrick Method에서 제외시켜준다.**

**2. minterm(row) – PI(column) 형식의 테이블을 생성한다.**

**3. minterm 의 수만큼 for문을 돌며, 이전 반복에 의해 minterm이 포함되지 않고 pi가 선택되지 않은 경우, minterm에 해당하는 pi를 sop에 추가한다**

**4. 최종 결과값 sop를 리턴한다.**

### 시스템 구조 및 설계도

|  |
| --- |
|  |

**(1) PI 테이블에서 모든 PI를 찾는다.**

**(2) 테이블에서 EPI를 찾고 제거한다.**

**=> 더이상 NEPI가 없으면 QUIT**

**(3) Column Dominance를 적용한다.**

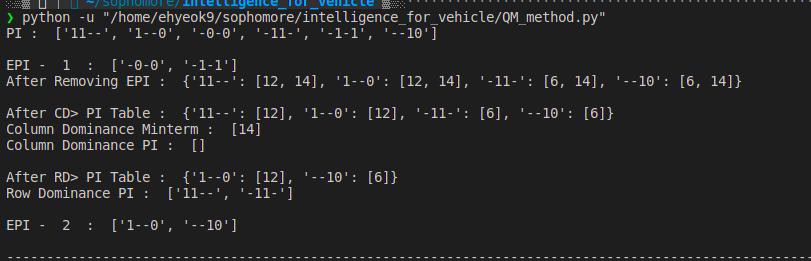
**(4) Row Dominance를 적용한다.**

**(5) (2)와 (3)을 통해 어떠한 최적화가 되었다면 (2)로, 되지 않았다면 Petrick method를 적용한다.**

### 결과물 목록

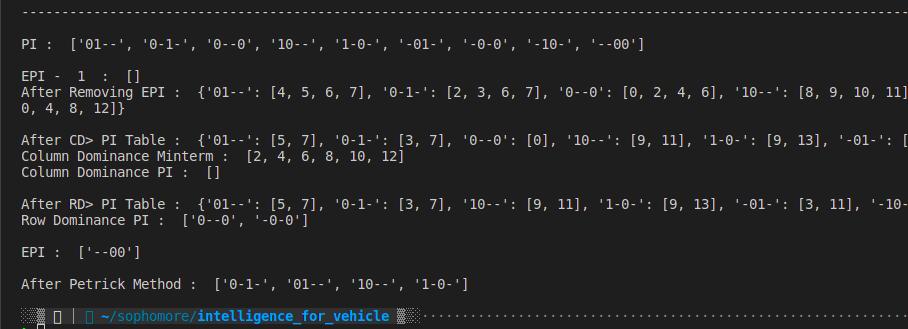
1번 테스트 케이스

|  |
| --- |
|  |



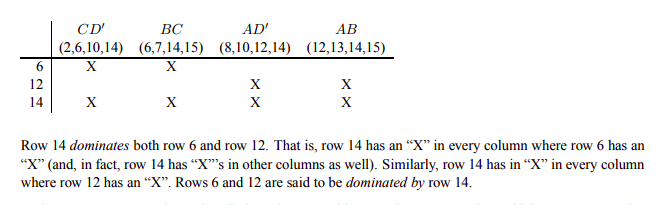
2번 테스트 케이스



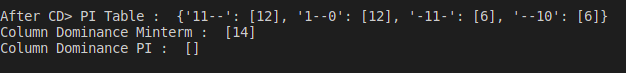


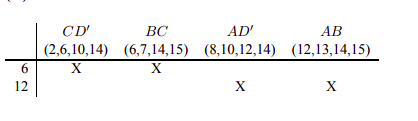
**- 강의 자료의 petrick 파트에 있는 Minterm을 예시로 활용하였다.**

**º 1번 테스트 케이스**



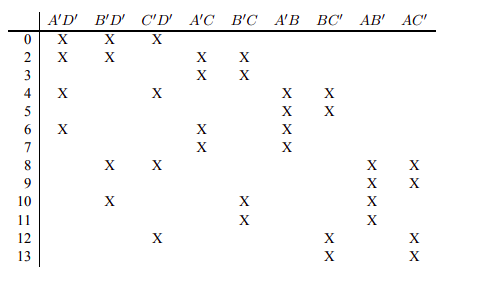
**1 ) Column dominance 후 minterm 14가 minterm 6,12를 지배하므로 minterm 14를 제거한다.**

****

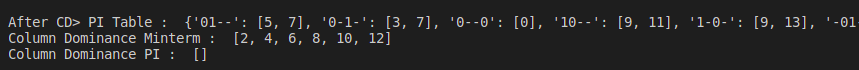
****

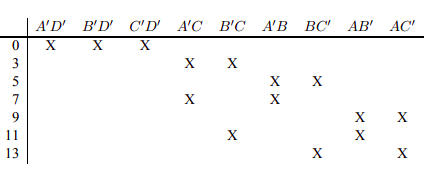
**2 ) ‘11--’과 ‘1–0’, ‘-11-’과 ‘--10’이 서로를 지배하는 관계(Interchagable)이므로 임의로 제거한다**

**º 1번 테스트 케이스**

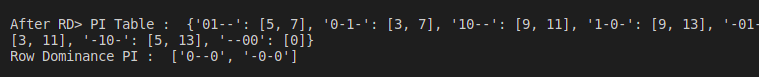


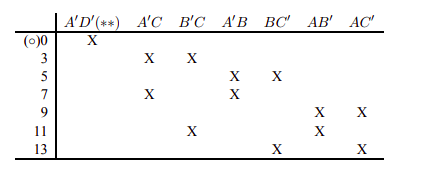
**1 ) Column Dominance : 2 → 3, 4 → 5, 6 →7, 8→ 9, 10 → 11, 12 → 13 이렇게 지배하는 - 지배당하는 구조가 형성된다. 지배하는 [ 2, 4, 6, 8, 10, 12 ] minterm을 제거한다.**

****

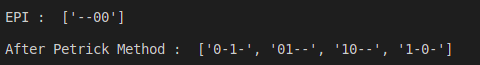
****

**2 ) Row Dominance : ‘--00’, ‘0–0’, ‘-0-0’이 서로를 지배하는 관계(Interchagable)이다. 임의로 하나의 pi만 남긴다.**

****

****

**3 ) Petrick Method : minterm 0에 해당하는 pi는 다음 단계에서 찾아야 할 epi 이므로 제외시켜준다. 나머지 minterm들을 통해 최소 논리곱의 합(SOP)를 얻는다.**

****

# 참고 문헌

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
| 1 | 파일 | https://cseweb.ucsd.edu/classes/su14/cse140-a/handouts/Quine.pdf |  |  |  |  |