

보드 생산 자동화를 위한 IC 정보의 호환성 Solution Compatibility Solution of IC Information for Board Production Automation

김귀정
건양대학교

Kim Gui-Jung
Konyang University

요약

본 논문은 IC 부품 정보를 이용하여 CAD 데이터를 PCB 보드 생산 공정에 적합한 데이터로 자동 변환함으로써 이기종 간의 호환성 문제를 해결하는 방법을 제안한다. 본 논문은 NC Programming 자동화로 작업능률을 극대화하는 것에 목적이 있으며, 서로 다른 Mounter 간의 NC Program Conversion(호환)을 이루고 Mounter Data & Line Balance 최적화로 생산성을 극대화시키는데 기여할 수 있다.

Abstract

This paper proposes the solution of compatibility problem between heterogeneous by converting automatically CAD data into proper data for the PCB board production process by using IC item information. This paper has a purpose of maximizing work efficiency with NC Programming automation and contributing to maximize productivity by NC program conversion of different Mounters and Mounter Data & Line Balance optimization.

I. 서론

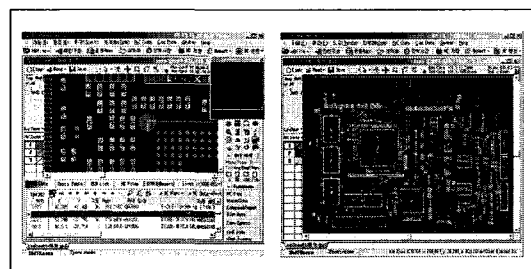
본 논문은 PCB(Printed Circuit Board) 보드 생산에 있어 설계 데이터를 공정 데이터로 변환하기 위한 수작업을 자동화함으로써 호환성 문제를 해결하는 방법을 제안한다. PCB 생산 공정은 CAD로 설계한 데이터를 인식하지 못한다. 따라서 많은 중소기업들은 수작업에 의한 데이터 변환을 수행하다보니 에러가 발생하고 정확도가 떨어져 불량 제품이 생산되는 경우가 빈번하여 많은 비용이 들고 있다.

본 논문에서는 PCB 조립 공정에서 CAD로 설계된 IC(Integrated Circuit) 부품별 데이터를 분석하여 데이터 호환을 위한 방법을 제시한다. 이는 CAD 설계자의 Human Error를 막기 위해 품목 명, 위치, 각도 데이터를 PCB 생산 공정에 적합한 데이터로 추가, 수정해 준다. IC 부품 정보에서 위치 정보를 기본 키로 하여 Join 연산과 각도 수정, 좌표 수정, Layer 분리를 통해 데이터를 변환한다. 그 결과 서로 다른 Mounter 간의 NC Program Conversion(호환)이 이루어지고, Mounter Data & Line Balance 최적화로 생산성이 극대화될 수 있다. 이로 인해 PCB 대량 생산이 가능해지고, 보다 정확한 PCB 생산과 높은 신뢰도를 얻을 수 있다.

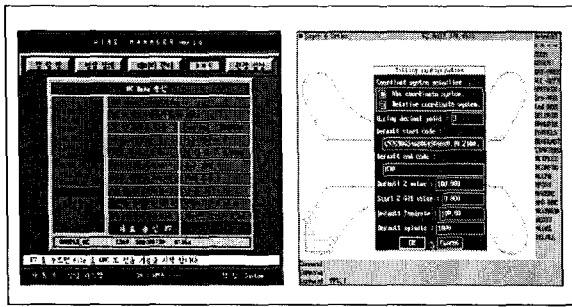
II. 관련 연구

현재 개발된 NC 소프트웨어 종류는 크게 두 가지로 구분할

수 있다. 첫 번째는 CAD와 연동하여 데이터변환을 할 수 있는 소프트웨어이다. 이는 CAD를 다룰 수 있다는 전제하에 소프트웨어를 사용할 수 있다. 그림 1은 CAD와 연동하여 데이터변환을 하는 소프트웨어를 보여준다. 이 소프트웨어는 그래픽 설계를 강조한다[1]. 따라서 그래픽 설계에 대한 전문적인 지식과 CAD에 대한 지식을 알고 있다면 편리하게 사용할 수 있다. 또한 그래픽으로 현재 설계된 PCB 제품을 확인할 수 있는 장점이 있다. 하지만 사용자가 그래픽 설계에 대한 전문적인 지식이 없다면 사용하기 불편하다. 현재 PCB 생산을 하는 중소기업에서는 이러한 전문 인력을 확보하기가 쉽지 않다. 두 번째는 명령어 위주의 소프트웨어이다[2]. 단순 NC 작성 소프트웨어로 개발되었기 때문에 빠른 시간에 데이터 변환을 하는 장점이 있다. 하지만 사용자는 명령어를 소프트웨어에 직접 입력해야하는 번거로움과 명령어를 확실히 인식하고 있어야 작동하는 불편함이 있다. 그림2는 명령어 위주의 NC 작성 소프트웨어이다.



▶▶ 그림 1. CAD와 연동하는 NC 작성 소프트웨어



▶▶ 그림 2. 명령어 위주의 NC 작성 소프트웨어

현재 개발된 NC 작성 소프트웨어는 고가의 장비임에도 불구하고 사용자가 쉽게 접근하기 어렵다. 소프트웨어가 특정 사용자를 위해 설계되었기 때문에 일반 중소기업에서는 이러한 요구 사항을 만족하지 못한다. 따라서 현재 많은 중소기업은 이러한 작업을 수작업으로 진행하는 것을 선호하고 있다. 하지만 아무리 숙달된 관리자라 하더라도 Human Error를 포함하기 때문에 신뢰성을 확보할 수 없다. 또한 수작업의 시간이 최소 8시간에서 검증을 위한 시간까지 포함하면 1-2일 정도가 더 소요된다. 따라서, 수작업의 비효율성을 해소하면서 사용자가 편리하게 사용할 수 있는 NC 작성 소프트웨어가 절실하게 필요하다[3].

본 논문에서 제시하는 시스템은 향후 NC 작성 소프트웨어의 요구사항을 충족하는 특징을 가진다. 데이터 변환 과정에서 데이터 입력이 반드시 필요하지 않는다면 자동으로 변환하도록 시스템을 설계한다. 그러면서 CAD 데이터를 PCB 공정에 맞는 최적화된 데이터로 변환하여 신뢰도를 높인다.

III. 데이터 변환

1. PCB 데이터 저장소

PCB 부품 정보 저장소는 부품 위치, 각도(X,Y값), 규격, 품목명, 구성 수량 등의 정보로 구성된다. 이 정보들은 CAD 데이터 중에 변환 과정에 필요한 정보만을 추출한 것이다. 저장소는 두 개의 Table을 가진다. IC 부품 저장소의 정보를 바탕으로 PCB 설비 장비가 인식할 수 있는 데이터들로 변환하게 되는 데이터 구조는 표 1과 같다. 'Table1'의 'Location_No'과 'Table2'의 'RefDes' 필드는 각각 위치 정보를 갖는다. 'Table1'의 'Q_ty' 필드는 앞서 데이터 추출에서 사용했던 'Location_No'의 위치 정보에 대한 개수를 의미한다. 그리고 'Table1'에서 나머지 필드는 Vendor에 대한 정보이다. 'Table2'의 'ParType'과 'PartDecal' 필드는 부품에 대한

상세 정보이다. 'Pins' 필드는 부품의 핀 수를 의미한다. 'Layer' 필드는 PCB 보드에 앞면 혹은 뒷면에 부품이 접합될 것인지에 대한 정보이며 PCB 생산 라인을 선택하는 자료로 사용된다. 'Orient' 필드는 부품의 각도이고 'X'와 'Y' 필드는 좌표를 의미한다. 나머지 필드는 부품에 대한 상세 정보를 포함한다.

[표 1] PCB 부품 저장소의 데이터 구조

Table	Field Name	Data Type	설명
1	Specification	String	저항
	Q_ty	long	수량
	Location_No	String	위치
	Approved Vendor 1_st	String	Vendor 정보
	Approved Vendor Parts No.1	String	Vendor 정보
	Approved Vendor 2_nd	String	Vendor 정보
	Approved Vendor Parts No.2	String	Vendor 정보
2	PartType	String	부품 정보
	RefDes	String	위치
	PartDecal	long	부품 정보
	Pins	long	핀수
	Layer	bool	앞, 뒤
	Orient	long	각도
	X	long	좌표
	Y	long	좌표
	SMD	bool	부품 정보
	Glued	bool	부품 정보

2. CAD 데이터 오류 검출 및 데이터 추출

오류 검출 알고리즘은 CAD 데이터 내의 에러 유형을 파악하고 이를 수정한다. CAD 데이터에서 품목명, 위치, 각도 값이 PCB 생산 공정에 이미 저장된 정보와 일치하는지 파악한다. 일치하지 않을 때 에러로 간주한다. 품목명은 CAD 데이터 중 자재관리시스템의 데이터베이스에 저장되지 않은 품목명이 발견되면 오류 메시지를 출력함과 동시에 새로운 품목명을 입력 받는다. CAD 데이터에서 원점은 PCB 보드의 가운데 부분으로 한다. 따라서 원점을 중심으로 하는 좌표(X,Y) 값이 70이상의 경우가 없다. 70이상의 경우가 발생할 경우 이를 오류로 간주한다. 각도는 0°, 90°, 180°, 270° 만이 존재하고, 그 외의 값이 발견되거나 값이 없을 경우 오류로 간주한다. 그림 3은 두 개의 Table에 저장된 CAD 데이터 중에서 부품명, 좌표, 각도에 따른 오류를 발견하는 알고리즘이다. 하나의 Table에서 filed 별로 오류를 발견하고 마지막 Recode까지 반복한다.

```

Specification, Location_No, 1_st, Parts_No_01, 2_nd,
Parts_No_02 save in Database Table1
PartType, RefDes, PartDecal, Pins, Layer, Orient, X, Y
save in Database Table2
FOR(Last Recode of Table 1)
  IF(Item Name of Material Management DB != Item
    name of Table1)
    Message Box
  ENDIF
Save Item Name in Database Table1
FOR(Last Recode of Table 2)
  IF(Orient of Table2 != 0 || Orient of Table2 != 90 ||
    Orient of Table2 != 180 || Orient of Table2 != 270)
    Message Box
  ENDIF
  IF(Coordinate of Table2 > 70 || Coordinate of Table
    == NULL)
    Message Box
  ENDIF
Save Orient, Coordinate in Database Table2

```

▶▶ 그림 3. 오류 검출 알고리즘

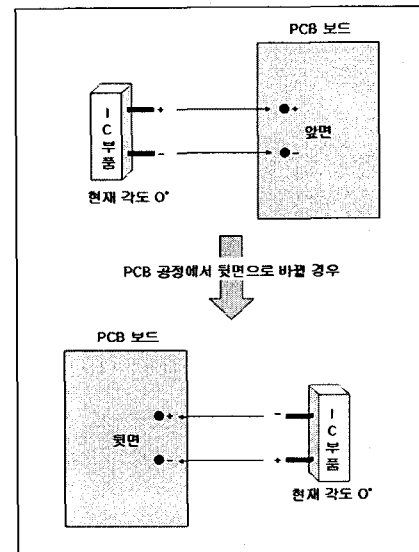
CAD 데이터는 설계자의 편의에 의해 설계되어 PCB 생산 공정과는 관계없는 불필요한 데이터를 가진다. 이런 불필요한 데이터는 PCB 생산 공정 분석을 통해 걸러 낸다. 분석된 결과에 따라 CAD 데이터 중 실질적으로 필요한 데이터만을 추출한다. 설계자의 편의에 의해 삭제된 일부 데이터들이 PCB 생산 공정에서는 유용한 정보 일 수 있다. 즉, 불필요한 데이터는 삭제하고 필요한 데이터는 추가하여 데이터를 추출한다. 이러한 단계를 거치게 되면 저장소에는 오류 없는 데이터를 저장하게 된다.

3. 부품 정보 변환

IC 부품에 대한 데이터와 CAD 데이터를 비교하여 부품 정보를 변환한다. 'Table1'의 'Location_No' 필드와 'Table2'의 'RefDes' 필드를 서로 비교하여 같은 데이터를 찾아 Join 연산을 수행한다. 'Location_No'와 'RefDes'의 값은 PCB 보드에 어떤 위치에 IC 부품을 조립할 것인지에 대한 정보를 가진다. 이 데이터를 기본키로 한다. 두 Table을 비교하여 같은 데이터가 있다면 'Table1'의 'Specification', 'Qty'와 'Table2'의 'Orient', 'X', 'Y', 'Layer'를 데이터베이스 내의 다른 Table에 저장한다. 물론 'Location_No'와 'RefDes'의 값은 Join이라는 하나의 필드에 저장한다. 이 값들은 PCB 보드에 어떤 위치에 어떤 부품이 들어갈 것인지를 알려주는 최소한의 정보이다.

CAD 데이터의 부품 각도인 'Orient'의 경우 CAD 설계자에 의해 임의의 각도 값이 주어질 가능성이 있다. 설계자는 PCB 보드와 상관없이 IC 부품의 표준 각도를 그대로 적용하기 때문이다. 따라서 PCB 보드에 따라 그 표준 각도의 변환하여 데이터베이스에 저장한다. IC 부품의 표준 각도는 0°, 90°, 180°, 270°이다. PCB 생산 공정에서 설계자의 생각과 달리 IC

부품이 PCB 보드에 앞면 혹은 뒷면에 접합되는데 이 Layer가 바뀔 수 있다. 그림 4를 보면 앞면에 부착할 것으로 예상한 설계자가 PCB 공정에서 뒷면에 부착하기로 하였다면 IC 부품의 +극과 -극이 서로 바뀌었기 때문이다. 따라서 Layer의 수정이 불가피한 상황이 발생할 경우 IC 부품의 각도가 0°는 180°로 바뀌고 90°는 270°로 바꾼다. 각도는 데이터베이스의 Join 연산 후 생성된 Table의 'Orient' 필드에 저장된다.

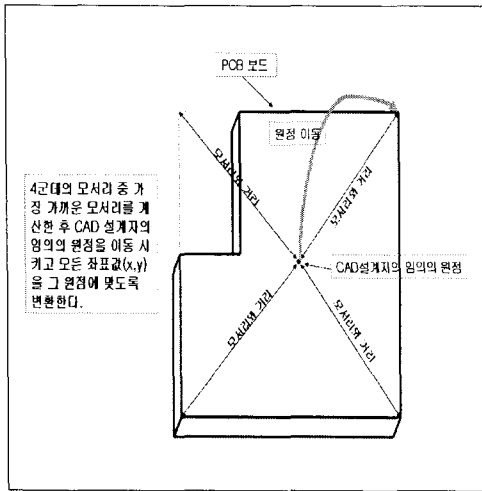


▶▶ 그림 4. IC 부품의 접합 Layer가 바뀔 경우

4. 좌표 수정

하나의 Table로 저장된 데이터에서 IC 부품별 데이터는 CAD 설계자의 임의의 원점을 중심으로 한 좌표(X,Y) 값을 가지고 있다[4]. CAD 데이터와 PCB 생산 공정에서 사용하는 좌표 값은 서로 다른 체계로 되어있다. 따라서 CAD 데이터의 좌표 값이 어떤 위치를 중심으로 설계되었는지 확인하고 PCB 생산 공정에 적합한 좌표 값으로 변경한다.

CAD 설계자는 좌표 값을 PCB 보드 중간 정도의 위치를 원점으로 정한다. 반면에 PCB 생산 공정에서는 PCB 보드 모서리 부분을 원점으로 사용한다. 따라서 CAD 설계자가 어떤 위치를 원점으로 정했는지 원점 알고리즘을 통해 원점을 찾는다. 이 원점에서 PCB 보드 네 군데의 모서리 중에서 가장 가까운 모서리를 찾는다. 네 모서리 중 가장 가까운 모서리를 원점으로 바꾼다. 이렇게 새로운 원점을 정한 후, IC 부품의 좌표 값을 새로운 원점을 중심으로 바꾼다. 그림 5는 CAD 설계자의 임의의 원점을 찾아 낸 후, 새로운 원점을 찾아내는 과정이다.

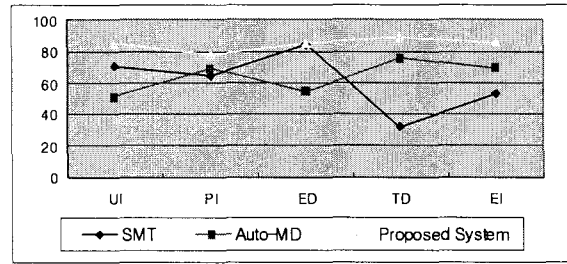


▶▶ 그림 5. 원점 수정

IV. 성능 평가

논문에서 제시한 방법과 기존의 데이터 변환 시스템을 비교한다. 비교 기준은 오백 개 이상의 부품 데이터를 갖는 CAD 데이터를 PCB 생산 공정에 적합한 데이터로 변환하는 과정으로 한다. 변환 과정은 총 35회 실시하였다. 비교 기준은 작업을 수행하는데 시스템이 얼마나 도움을 주는가의 정도와 사용하는데 얼마나 쉽고 편한가에 따라 소프트웨어 시스템의 질을 평가한다. 평가 목록으로 효율성의 증가(UI), 생산력의 개선(PI), 에러의 감소(ED), 훈련의 감소(TD), 사용자 수용의 증가(EI)로 한다[5][6]. 제시한 항목은 만족도를 100으로 하고 총 8명의 사용자에게 질의하였다

그림 6은 기존 시스템과 본 논문이 제시한 방법을 비교한 그래프이다. SMT의 경우 UI, ED에 높은 점수를 보이지만 PI, ED, TD에 낮은 점수를 보였다. 특히 ED와 TD에서 매우 낮은 평가를 받았는데 이는 CAD를 사용하지 못하는 사용자가 많았음을 의미한다. Auto-MD는 PI, TD, EI에서 높은 점수이었지만 UI, ED에서 낮은 점수를 보였다. 이는 명령어 위주의 복잡한 사용 방법으로 평가된다. 그러나 본 논문에서 제시한 방법은 전체적으로 SMT와 Auto-MD보다 높은 점수를 얻었다. 이는 사용자에게 편리한 인터페이스를 제공하면서도 시스템적인 오류가 없음을 의미한다.



▶▶ 그림 6. 기존 시스템과 비교

V. 결론

본 논문은 PCB 보드 생산 자동화를 위하여 IC 부품 정보를 이용하여 CAD 데이터를 PCB 생산 공정에 적합한 데이터로 변환하는 것이다. CAD 데이터에서 PCB 생산 공정에 필요한 데이터를 추출하고 위치정보와 누락된 정보를 확인하여 자동으로 수정한다. CAD 데이터 내의 오류를 검출하고 제거한 후, 데이터를 PCB 생산 공정에 사용하는 데이터로 변환할 수 있도록 하였다. 이에 따라 본 논문은 IC 부품 정보를 활용하여 이기종 간의 데이터 호환성 문제를 효율적으로 해결할 수 있었다. 따라서, 서로 다른 Mounter간의 NC Program Conversion이 이루어지고, Human Error들이 사전에 제거될 수 있다. 또한, PCB의 대량 생산이 가능해지고, 보다 정확한 PCB 생산과 높은 신뢰도를 얻을 수 있을 것이다.

참고 문헌

- [1] <http://www.smtkorea.co.kr>
- [2] <http://www.changil21c.co.kr/>
- [3] 김경수, "PCB 회로 설계", 홍릉과학출판사, 2004.
- [4] Park S.S and Sohn J.H, "Efficient operation of a surface mounting machine with a multi thread turret", International Journal of Production Research", pp.1131-1143, 1996.
- [5] 송영재, "객체지향모델링과 CBD 중심 소프트웨어 공학", 이한출판사, 2004.
- [6] M.lea, "Evaluation User Interface Design, user Interface Design for Computer System", Halstead Press, 1988. pp.1131-1143, 1996.