

LP Algorithm을 이용한 반도체 생산 계획의 도출

박동식, 이지형, 유관호, 이철기
성균관대학교 컴퓨터공학과

A Study of Making Semiconductor Production Plan using LP Algorithm

Dong-Sik Park, Jee-Hyong Lee, Kwan-Ho Yu, Chil-Gee Lee

Abstract

To make production and equipment investment plans in semiconductor Line, implementation of many variables is needed. But these factors could bring many changes and the result is hard to predict. Because prediction is hard, it is hard to make a standard. So this project established Semiconductor production plans using LP Algorithm to satisfy all the conditions from the factors and came up with thesis on reasonable and standardized process.

I. 서론

반도체 산업은 21세기 정보화 사회를 선도하는 핵심 부품 산업으로 국가의 첨단 산업 발전을 촉진하는 주력 산업이다. 반도체 산업은 첨단 핵심 부품산업으로써, Timing산업, 고부가가치, 고성장, 고 위험, 기술 집약적 특징을 갖는다. 따라서 시장상황을 정확히 예측하고 그 상황에 적절히 대응할 수 있는 생산 계획을 세우는 일은 매우 중요하다. 그러나 생산 계획을 세우는데 충족되어야 하는 제약조건은 매우 다양하여 이것을 모두 만족시키며 생산계획을 수립하는 것은 매우 어렵다. 그동안은 사람의 주관적인 생각으로 하다보니 시간이 너무 많이 걸려서 자동화하는 방법이 필요하다. 이에 본 과제에서는 LP Algorithm을 이용하여 여러 변수의 제

약조건을 만족시키면서, 합리적인 반도체 생산계획을 수립하는 방법을 제안한다.

본 논문 구성은 다음과 같다. 2장에서는 반도체 생산 계획에 대해 설명하고, 3장에서는 LP Algorithm을 이용해 생산계획을 세우는 것에 대해서 설명하였다. 마지막 4장에서는 결론 및 추후 연구 방향을 제시하였다.

II. 반도체 생산계획

반도체를 라인에서 생산하기 위한 계획을 수립하는 데에는 다음과 같은 절차를 따르게 된다.

2.1 판매계획 수립

해당 시기에 어느 제품을 얼마만큼 판매할 것인지 계획을 수립하는 것을 말한다. 판매계획은 특정제품의 수요전망과, 당사 매출 및 시장점유율 전망, 제품별 판매 단가를 참고로 하여 수립하게 된다.

2.2 Design Rule(이하 DR)

반도체의 설계과정에서 공정능력과 설정된 제조방법의 한계성을 고려하여 반드시 지켜져야만 하는 설계규칙으로서 각 기능 부위 및 기능부위 간의 물리적 거리가 그 내용으로 포함한다. 같은 제품이라도 여러가지 DR을 사용해서 제조할 수 있고 제품에 따라 적용할 수 있는 DR이 각각 다르다. 또한 DR에 따라서 웨이퍼당 산출되는 칩의 갯수가 틀리다.

2.3 라인별 생산가능용량(Capacity) 산정

각 라인은 생산가능용량을 가지고 있다. 하지만, 보유하고 있는 설비에 따라 생산에 사용할 수 있는 DR이 정해져 있다. 따라서 생산가능용량을 산정하기 위해서는 라인별 DR별 생산가능용량으로 산정해야 한다.

예를 들어 생산라인 두개가 있고 사용가능한 DR이 3개가 있을때, 1라인에서는 DR1, DR2를 사용할 수 있고 2라인에서는 DR2, DR3를 사용할 수 있다면, 다음표와 같이 각 라인별 DR별 용량을 산정할 수 있다.

라인	DR	Capacity
1라인	DR1	20000
	DR2	25000
2라인	DR2	15000
	DR3	30000

2.4 제품별 DR별 개발계획

제품별 DR의 개발 시기를 파악하여 어느 시기에 라인에 투입될 수 있는지 파악하는 것이다.

2.5 생산비중 산출

생산비중 산출이란 위의 4가지 데이터를 참조로 하여, 특정 제품, 특정 DR의 생산비중을 산출하는 것이다. 생산비중이란 제품별 DR의 비율을 말한다.

예를들어, DR1과 DR2, DR3를 적용시켜 생산할 수 있는 A 제품이 있다고 하자. 그렇다면 다음표와 같이 제품별 DR의 생산비중을 산출할 수 있고, 그 합은 100%가 되어야 한다.

제품명	DR	생산비중
A	DR1	30
	DR2	20
	DR3	50
	합	100

생산비중을 산출하는데 있어서도 여러 알고리즘이 쓰일 수 있는데, 본 논문에서는 패턴기반의 알고리즘으로 계산된 생산비중을 근거로 하여 라인 분배를 실시하였다.

2.6 라인 분배

위의 여러 요소들을 산출하여 라인의 DR별 생산가능 용량에 따라 적절하게 분배를 해 주어야 한다. 이때 생산가능용량이 여유가 있다고 무조건적으로 분배를 하게 되면, 여러 문제들이 발생을 하게 된다.

다음과 같은 간단한 예를 보자. 2005년 1분기에 다음과 같은 양의 제품을 생산하여야 한다고 하자.

제품	DR	필요량
A	DR1	20000
	DR2	10000
	DR3	10000
B	DR2	15000
	DR3	20000

라인별 DR별 생산가능용량은 다음과 같다고 하자.

라인	DR	Capacity
1라인	DR1	20000
	DR2	25000
2라인	DR2	15000
	DR3	30000

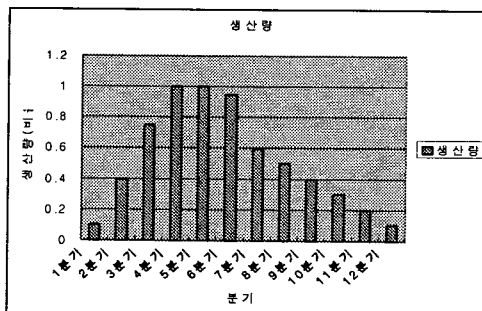
그렇다면 다음과 같이 배분을 하면 웨이퍼 생산량 필요량을 라인별 생산가능용량내에서 생산할 수 있다.

라인	제품	DR	Capacity
1라인	A	DR1	20000
	A	DR2	5000
	B	DR2	5000
2라인	A	DR2	5000
	B	DR2	10000
	A	DR3	10000
	B	DR3	20000

위와 같은 작업을 각 분기별로 수행하여 생산라인 운영계획을 세우게 된다. 하지만 위의 작업에는 여러가지 어려움이 따르게 된다. 첫번째로는 라인, 제품및 DR의 종류가 수십 내지 수백가지 정도가 된다. 이러한 이러한 변수들이 서로 연관되어 영향을 주고 받아서 한가지 변수가 변동되면, 다른 많은 변수들도 변동이 되어야 한다는 점이다.

두번째로는 각 라인별로 생산해야 하는 제품별 DR별 생산량이 각 분기별로 연속성을 가지고 있어야 한다는 점이다. 각 제품은 그냥 생산만 하면 되는 것이 아니라, 라인의 설비 상황을 감안해야 한다. 반도체 라인에서, 어느 제품의 생산량을 증가시키거나 감소시키려면 설비의 업그레이드나 재배치가 필요한데, 그러한 것을 감안하지 않고 무리한 생산량의 증가나 감소를 시킬 경우 문제가 발생하게 된다. 예를 들어, 이번 분기에 제품 500개를 생산하는 계획을 세웠다면, 다음분기에는 +800에서 -200 범위내에서 생산 계획을 세워야 무리가 없을 것이다. 일반적으로 분기의 흐름에 따라서, 라인별 DR별 제품별 생산량은 다음과 같은 패턴으로 생산해 주는 것이 이상적이다.

이외에도, 위의 예에서는 각 제품의 필요량은 라인



<그림 1> 생산자료 패턴

의 생산가능용량내에서 생산할 수 있었지만 현실적으로는 그렇지 않을수도 있다. 예를 들면, 라인을 통틀어서 DR1의 생산가능용량이 20000인데 생산해야 할 양은 10000밖에 안 되고, DR2의 생산가능용량이 35000인데, 생산해야 할 양이 50000일 경우, 문제가 발생하게 된다. 이러한 경우에는 라인을 신증설 하거나, 혹은 제품을 변동시킬수 있도록 정보를 제공해야 한다.

위와 같은 여러가지 요인들 때문에 계획을 세우는 데 있어서 전문가도 상당한 시일이 걸리는 어려운 작업이다. 이에 본 과제에서는 라인 분배 과정을 LP Algorithm을 이용하여 여러가지 제약 조건을 만족시키면서, 최적인 방안을 찾도록 하였다.

III. LP에 의한 라인 분배

3.1 LP Algorithm

LP Algorithm,은 수리계획법 중 가장 기본적이고 널리 이용되는 기법으로 기업 등의 조직에서 최적 의사결정이 필요한 경우 기본적으로 고려되는 분석도구이다. LP Algorithm을 적용하기 위해 모델링 한 것을 LP 모델이라고 한다. LP 모델은 의사결정상황을 1차식의 함수로 표시한 것으로서 몇 가지 가정을 필요로 하는데 일단 모형화가 이루어지면 Simplex Algorithm에 의해 쉽게 해를 구할수 있다.

Simplex Algorithm,은 LP Algorithm중에서 가장 대표적인 수치 계산법이다. 기본 형태는 다음과 같다.

$$\begin{aligned} & \text{Maximize(또는 minimize) } f(x) \quad \Leftrightarrow \text{목표함수} \\ & x_1 + x_2 + \dots + x_n \leq Q_1 \\ \text{Subject to } & x_1 + x_2 + \dots + x_n = Q_2 \quad \Leftrightarrow \text{제한조건} \\ & x_1, x_2, \dots, x_n \geq 0 \end{aligned}$$

위의 형태로 표현된 모델링 결과에 대해 Simplex 테이블을 작성하고, 알고리즘 순서를 따라서 최적의 해를 구하게 된다. 이론적으로는 변수의 개수에 따라 계산량은 지수함수적으로 늘어나지만, 다른 LP Algorithm에 비해 안정적이며, 효율적이라는 장점을 갖고 있다.

그러나 LP 모델의 가정은 제한 조건 설정에서 비교적 정적인 측면이 있기 때문에, 이를 보완하기 위한 방법이 필요하다. 본 과제에서는 분기별 연속성, 생산 라인별/제품별/DR별 우선순위를 고려한 배분 문제를 해결해야 하므로, 비선형적인 제한 조건을 추가로 해결하는 방안을 모색해야 한다.

3.2 LP Algorithm 구축

LP Algorithm을 위한 변수 설정을 다음과 같이 수립하였다.

$$X_{\text{quarter, line, prod, dr}}$$

X : 생산할 웨이퍼 양
 quarter : 생산할 분기
 line : 생산할 라인
 prod : 생산할 제품명
 dr : 생산할 DR

예를 들어, $X_{1, A, m, 1}$ 은 1분기에 A라인에서 m이러한 제품을 위해 DR1으로 생산해야 할 웨이퍼 양을 의미한다.

여러 상황을 고려하여 다음과 같은 4가지 방정식을 수립하였다.

각 분기별, 모든 라인에 있어서 각각의 라인에서 생산할 웨이퍼 양은 라인별 생산가능용량보다 작거나 같아야 한다.

$$\begin{aligned} \sum_{\text{for all } i} (X_{\text{quarter}(1), \text{line}(1), i, \text{dr}(1)}) & \leq \text{LineCapa}(\text{quarter}(1), \text{line}(1), \text{dr}(1)) \\ & \dots \\ \sum_{\text{for all } i} (X_{\text{quarter}(n), \text{line}(k), i, \text{dr}(u)}) & \leq \text{LineCapa}(\text{quarter}(n), \text{line}(k), \text{dr}(u)) \end{aligned} \quad \text{<식 1>}$$

또한, 모든 제품에 있어서 각각의 합은 웨이퍼 제품 필요량보다 작거나 같아야 한다.

$$\begin{aligned} \sum_{\text{for all } i} (X_{\text{quarter}(1), \text{line}(1), i, \text{dr}(1)}) & \leq \text{WaferProdQuantity}(\text{quarter}(1), \text{line}(1), \text{dr}(1)) \\ & \dots \\ \sum_{\text{for all } i} (X_{\text{quarter}(n), \text{line}(k), i, \text{dr}(u)}) & \leq \text{WaferProdQuantity}(\text{quarter}(1), \text{line}(1), \text{dr}(1)) \end{aligned} \quad \text{<식 2>}$$

모든 DR에 있어서 각각의 합은 웨이퍼 DR 필요량보다 작거나 같아야 한다.

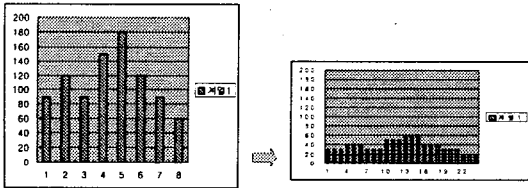
$$\begin{aligned} \sum_{\text{for all } i} (X_{\text{quarter}(1), \text{line}(1), \text{prod}(1), i}) & \leq \text{WaferDRQuantity}(\text{quarter}(1), \text{line}(1), \text{prod}(1)) \\ & \dots \\ \sum_{\text{for all } i} (X_{\text{quarter}(n), \text{line}(k), \text{prod}(u), i}) & \leq \text{WaferDRQuantity}(\text{quarter}(1), \text{line}(1), \text{prod}(1)) \end{aligned} \quad \text{<식 3>}$$

또한, 분기별로 연속성을 보장해 주기 위하여 다음과 같은 방정식을 수립하였다.

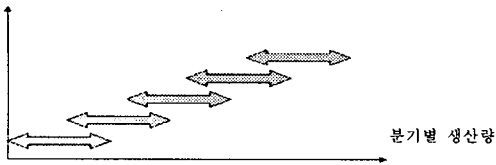
$$X_{i-1, \text{line, prod. dr}} + e(i-1) \leq X_{i, \text{line, prod. dr}} \leq X_{i, \text{line, prod. dr}} + e(i+1)$$

e(i) : 각 분기별 헤이퍼 변동 가용량
<식 4>

위 <식 4> 방정식을 적용했을 경우, 때로는 해답이 안 나올수 있다. 따라서 해답을 내기 위해서 급격한 변동도 허용해 줄 필요가 있다. 따라서 <그림 1>과 같은 생산량 패턴이 안 나오고, 급격한 변동이 있는 생산량 패턴이 나올 수 있다. 따라서 부드러운 변동성을 보여주는 생산량 패턴을 구하기 위하여 분기별로 되어 있는 간격을 한 달 간격으로 세분화 하여 계산을 하였다. 예를 들어, 1분기에 900장을 생산한다고 하면, 1월달 300장, 2월달 300장, 3월달 300장으로 나누었다. 이렇게 나누어서 계산을 하게 되면, 변동성도 나누어지는 효과를 얻게 되어, 결국 분기별로 봤을 때는, 변동성이 이전 방식보다 현저히 낮아지게 된다.



위와 같은 작업을 하여 수년치의 작업을 묶어서 실행했을 경우, 비록 부드러운 변동성을 보여주었지만, 계산 시간이 너무 많이 걸리게 되었다. 따라서, 필요한 모든 구간을 묶어서 계산하지 않고, 3개 분기씩을 엮어서 계산을 하였다. 이론적으로는 연속성이 깨질 위험이 더 커지지만, 실질적으로는 거의 변화가 보이지 않고, 계산 시간을 빠르게 할 수 있었다.



4. 결론 및 추후 연구

반도체 생산계획을 효율적이고 객관적으로 수립하는 것은 반도체 생산 / 판매에 있어서 매우 중요하다. 본 논문에서는 LP Algorithm을 이용하여 반도체 생산계획을 수립하였다. 자동화된 절차를 이용함으로써 오류를 최대한 방지하고, 10일 이상의 시간이 걸리던 것을 3시간안으로 처리할 수 있도록 할 수 있었다.

본 논문에서는 각 분기별로 이미 정해진 라인 생산가능용량을 이용해서 생산량을 산출하도록 하였다. 라인 생산가능용량은 라인 신, 증설 계획도 염두에 두어야만 하기에, 여기서는 다루지 않았다. 향후 생산계획을 이용하여 라인 신, 증설 계획까지도 수립하는 과제를 수행하도록 하겠다.

참고문헌

- [1] Wein, L.M., On the relationship Between Yield and Cycle time Semiconductor Wafer Fabrication, "IEEE transactions on semiconductor Manufacturing, VOL.5, 1992
- [2] B.K. CHOI and B.H. KIM, "MES (manufacturing execution system) architecture for FMS compatible to ERP(enterprise planning system)", INT. J. Computer Integrated Manufacturing, 2002, VOL. 15, NO. 3, 274-284
- [3] SEMI E53-1296 Event Reporting , SEMI Standard 0200
- [4] Introduction to Linear and Nonlinear Programming, D.G. Luenberger, Addison-Wesley, 1973.
- [5] Stanley Middleman, K.Hochberg , Process Engineering Analysis in Semiconductor Device Fabrication , Mc Graw-Hill, Inc.
- [6] <http://www.semi.org/>
- [7] <http://www.selete.co.jp/>