

재작업이 존재하는 반도체 제조공정을 위한 실시간 작업투입 알고리즘

신 현 준[†]

[†]상명대학교 경영공학과

A Real-Time Dispatching Algorithm for a Semiconductor Manufacture Process with Rework

Hyun Joon Shin[†]

[†]Department of Management Engineering, Sangmyung University

ABSTRACT

In case of high-tech process industries such as semiconductor and TFT-LCD manufactures, fault of a virtually finished product that is value-added one, since it has gone throughout the most of processes, may give rise to quality cost nearly amount to its selling price and can be a main cause that decreases the efficiency of manufacturing process. This paper proposes a real-time dispatching algorithm for semiconductor manufacturing process with rework. In order to evaluate the proposed algorithm, this paper examines the performance of the proposed method by comparing it with that of the existing dispatching algorithms, based on various experimental data.

Key Words : Real-Time, Dispatching, Quality, Production scheduling, Parallelized flow-lines

1. 서 론

재작업(rework)은 반도체나 박막트랜지스터 액정 표시장치(TFT-LCD)와 같은 하이테크 산업에서 제조공정의 효율성을 가늠하는 중요한 이슈가 되고 있고, 하이테크 산업이 위치하고 있는 시장은 제품수명 주기상 성숙기에 자리잡고 있다. 따라서 전 세계의 제조업체들은 생산성 향상 및 제조원가 절감을 목표로 치열한 경쟁을 벌이고 있다. 더욱이 이들은 장치산업으로서 시설에 대한 투자비용이 크기 때문에 설비 투자 여부를 결정할 때 위험관리 측면에서 신중하게 판단해야 하는 어려움이 있다. 이렇듯 치열한 경쟁과 투자 위험성이 존재하는 환경에서 경쟁력을 갖추려면 제품의 품질도 향상시켜야 하며 동시에 고객의 요구사항도 절대적으로 만족시켜야 한다[1,2].

반도체 공정에서 가공을 위해 투입되는 기계에 따라서 제품의 제조사양(specification)은 다양하게 변할 수 있다. 즉 어떤 기계에서 가공되느냐에 따라서 때로는

원하는 제조사양을 만족시키는 제품이 나올 수도 있고 그렇지 않은 경우가 발생할 수도 있는 것이다. 제품에 요구 되는 특정 사양을 만족하지 못할 경우 그 제품은 사용이 불가능하기 때문에 재작업을 통하여 재가공을 하거나 아니면 버려지게 된다. 이는 재작업에 따른 가공시간과 그에 따른 비용이 추가적으로 발생하는 것을 초래하기 때문에 생산 계획을 수립할 경우 최소한의 재작업이 발생하도록 충분히 고려를 해야 한다.

이와 함께 제조업체가 경쟁력을 갖추기 위해 필요한 것은 고객의 요구사항을 만족시키는 것이고, 고객의 요구사항은 크게 두 부분으로 요약될 수 있다. 그 하나는 납기 내 적시 인도이고 또 하나는 품질이다. 가공한 제품의 품질이 일정 기준에 미치지 못하게 된다면 이는 불량관정을 받아서 폐기되거나 재작업을 통해 양품을 만들어야 하기 때문에 추가 발생하는 직접비도 무시할 수 없다. 따라서 납기준수 및 재작업 최소화를 동시에 고려하는 효율적인 생산관리 기법의 개발이 필요하다. 따라서 본 논문에서는 재작업을 고려한 작업투입(dispatching) 방법을 통해 품질비용을 최소화함과 동시에 납기지연을 최소화하는 방안을 제시하고자 한다.

[†]E-mail : hjshin@smu.ac.kr

2. 선행연구

재작업은 반도체 산업을 포함하여 많은 제조 산업에서 중요하게 고려되는 사항으로, 주로 비용의 최소화를 위해 최적의 생산배치크기나 재작업시기를 결정하는 문제 또는 최적의 재작업 정책을 결정하는 것에 대한 연구가 진행되어 왔다[3]. Liu는 단일제품 생산 환경 하에서 단위시간당 평균이익을 최대화시키는 최적의 생산배치크기를 결정하는 방법을 제시하였고[4], So & Tang은 정상(Regular)제품과 재가공 제품을 번갈아 가공하는 공정에서 Semi Markov Decision Process를 이용하여 평균운용비용을 최소화 시키는 최적의 운용정책(operating policy)을 결정하는 문제를 다루었다[5]. Lee 등은 반도체 MBT 공정에서 재작업으로 발생하는 제조원가 절감이익과 추가로 발생하는 재가공비용 및 재고비용 사이의 trade off를 고려한 재작업 투입정책을 제안하였다[6].

재작업 정책을 dispatching rule과 결합한 연구도 있었다. Zargar는 ‘mother lot’과 ‘child lot’의 개념을 도입하고 그들을 처리하는 방법에 따라 4가지 재작업 정책을 제시하였다. 또한 실험을 통해 재작업 정책들이 cycle time에 미치는 영향에 관한 결과를 보여주었다[7]. Sha 등은 반도체 포토공정의 dispatching을 통해 전체 공정이 운용된다고 보고 고객의 납기를 만족시키고 동시에 높은 제품품질을 얻기 위해서는 재작업 전략과 dispatching 규칙을 적절하게 사용해야 한다고 언급했다 [8]. 이 때 세 가지의 재작업 전략과 세 가지 dispatching 규칙의 조합으로 시뮬레이션을 수행하였다.

이상과 같이 기존 연구들을 고찰해본 결과 재작업 확률을 실시간dispatching 알고리즘에 직접 반영하여 고안한 시도는 극히 드문 실정이며, 이는 일반 제조현장에서 재작업과 관련한 데이터를 체계적으로 취합하여 작업lot와 기계간의 세밀한 재작업 발생 데이터를 정립하지 못하였거나 제조실행시스템(manufacturing execution system)이 도입되었더라도 수율(yield) 데이터를 활용하기 위한 접근들이 미비했던 것으로 보인다. 일부 재작업을 고려한 연구들 또한 기존 여러 dispatching 규칙들을 상황에 따라 전략적으로 조합하여 사용하는 방식들이 대부분이며, dispatching 규칙의 성능을 향상시키기 위해 메타휴리스틱 등의 탐색알고리즘 또는 신경망 기법과 함께 사용하려는 시도가 있었으나 이는 자동으로 제어되는 하이테크 제조현장에서 재작업 발생 확률의 변화에 따라 작업의 dispatching policy를 동적으로 변화시켜가며 실시간적으로 신속하게 적용하기에는 어려움이 따른다.

본 논문은 다음과 같이 구성되어 있다. 3장에서는 본 논문의 대상이 되는 재작업이 존재하는 반도체 제조공정을 위한 작업투입 스케줄링 문제에 대해서 설명하고 4장에서는 본 논문이 제안한 작업투입 알고리즘에 대하여 기술하고 5장에서는 다양한 실험계획 하에서 실행된 실험 결과 및 분석을, 마지막으로 6장에서 결론 및 추후 연구 과제를 제시한다.

3. 문제정의

본 연구에서는 평균지연시간(mean tardiness; 이하 MT)을 최소화하기 위해 재작업 가능성이 존재하는 상황에서 N 개의 작업을 $M(m=1, \dots, M)$ 개의 병렬라인 형태의 기계에 투입하는 알고리즘을 제시한다. N 개의 작업은 같은 제품타입(product type)을 갖는 C 개의 작업군으로 나뉘고, 이 때 작업군 J_c 를 제품타입이 c 인 작업들의 집합이라 한다. 작업군 J_c 에 속한 작업들의 인덱스를 $j(j=1, \dots, N_c)$ 라고 했을 때, 이들 작업 j 는 각각 납기(due date) d_j 와 작업투입시점(release time) r_j 를 갖는다. 각 작업 j 의 가공시간(processing time) p_j 는 작업들이 기계에서 처리되는 순서 및 가공 기계와는 독립적으로 주어지지만 작업준비시간은 작업들이 가공되는 순서와 속한 작업군에 따라 달라지는 순서의존적인 준비시간(sequence dependent setup times)을 갖는다. 여기서 순서의존적인 작업준비시간이란 작업군 J_c 에 속한 작업 i 의 가공을 마친 후 작업군 J_c 에 속한 작업 j 를 준비하는 데 소요되는 시간을 $s_{c,c'}$ 라 할 때, $s_{c,c'} \neq s_{c,c'}$ 인 성질을 갖는 작업준비시간을 의미한다. 물론 작업 i 와 j 가 같은 제품군에 속한다면 $s_{c,c'}=0$ 이다. 이 때 작업군 J_c 에 속한 작업 j 가 기계 m 에서 가공을 한 후 불량 판정을 받아서 재가공을 할 확률은 P_{cm} 이다.

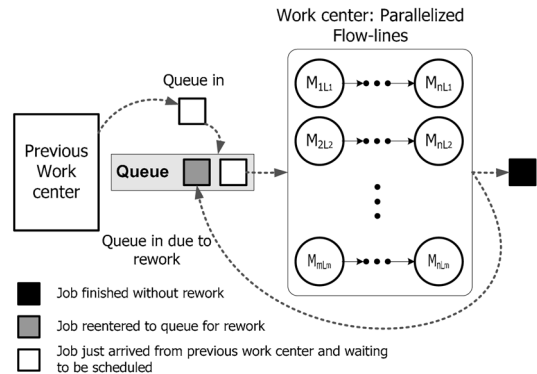


Fig. 1. System configuration.

의사결정 t 시점인 시점에 기계 m 이 가공 중이던 작업 $i(i \in J_c)$ 의 가공을 완료한 후 유휴(idle)상태가 되어 투입작업으로 작업 $j(j \in J_c)$ 가 선택되었다고 하자. 현재 재작업 확률이 존재하기 때문에 작업 j 는 기계 m 에서 가공을 완료한 후 다시 가공을 해야만 하는 상황이 발생할 수도 있다. 만일 작업 j 가 기계 m 에서 가공을 완료한 후 재작업 판정을 받지 않을 경우에는 작업준비시간과 가공시간의 합인 $s_{c'c} + p_j$ 만큼의 시간이 소요되지만 재작업 판정을 받을 경우에는 $s_{c'c} + p_j$ 뿐만 아니라 초기화 및 재가공에 필요한 시간이 더 소요된다. 따라서 의사결정 시점마다 납기 지연 최소화와 재작업 발생 가능성의 최소화를 동시에 고려하는 스케줄링이 필요하다. 전체 문제상황을 도식화하여 설명하면 Fig. 1와 같고, 본 논문에서는 이와 같이 정의되는 문제에 대한 솔루션으로 재작업을 고려한 실시간 작업투입 알고리즘을 제안하고자 한다.

4. 제안 알고리즘

각 제품 타입마다 어떤 기계에서 가공했느냐에 따라 서로 다른 재작업 가능성이 존재한다면 작업들은 재작업 가능성이 작은 기계에서 가공을 하는 것이 납기 준수의 측면이나 재작업 시 추가되는 비용 측면에서나 바람직할 것이다. 그럼에도 불구하고 모든 작업이 재작업 가능성이 작은 기계에서만 가공한다면 특정 기계에만 작업들이 몰리는 현상이 발생하게 되고 이는 재작업에 따른 지연손실 및 품질비용은 줄이지만 자신의 차례를 얻기 위하여 기계 앞에서 한없이 가공을 기다려야 하는 부작용이 발생하게 된다. 여기서 납기 만족과 재작업 최소화 간의 상충관계(trade-off)가 발생하므로 이 두 요인을 효과적으로 조절할 필요가 있다. 따라서 본 논문의 핵심은 이러한 상충관계를 절충함과 동시에 반도체 제조공정에서 요구되는 신속한 공정제어를 위한 실시간 작업투입 방안을 개발하는 것이다.

스케줄링 시스템에 많이 사용되는 실시간 dispatching 규칙에는 EDD(earliest due-date), ATCS(apparent tardiness cost with setup), CR(critical ratio) 등의 규칙이 있으나, 이러한 dispatching 규칙들은 재작업 특성을 반영하지 않고 있어 그대로 사용하기에 부적합하다. 따라서 본 논문에서는 기존 CR 규칙에 재작업 특성을 반영한 CRR(critical ratio with rework) 규칙을 고안하여 사용하도록 한다. CRR 규칙은 재작업 확률과 가공시간 및 셋업시간을 통해 얻어지는 가중치를 고려함으로써 대기행렬(queue)의 부하감소와 워크센터의 산출량(throughput) 증가를 통해 고객 납기만족을 목표로 한다.

Queue에서 대기하고 있는 모든 작업들을 대상으로 병렬라인 상의 임의의 기계(m)가 유휴상태(idle)가 되면 그 시점을 의사결정 시점 t 라 한다. 이 때 CRR 규칙은 식(1)과 같이 우선순위 함수($CRR_j(t,i)$)의 값을 구하고 가장 작은 값을 갖는 작업을 해당 기계(m)에 투입하도록 결정한다.

$$CRR_j(t,i) = CR_j(t,i) \times weight_j(i) \tag{1}$$

여기서

$$CR_j(t,i) = (d_j - t) / (p_j + s_{c'c}),$$

$$weight_j(i) =$$

$$\begin{cases} 1 & , \text{if job } j \text{ is new arrived one} \\ (p_j + \bar{s}_{c'c}) / (\hat{p}_j + \hat{s}_{c'c}) & , \text{if job } j \text{ is reentered for rework} \end{cases}$$

i 유휴상태가 된 시점(t) 바로 이전에 기계 m 에서 완료된 작업의 인덱스,

c' 작업 j 가 속한 job class,

c 작업 j 가 속한 job class,

$s_{c'c}$ 작업 i 가 속한 job class(c') 가공완료 후, 작업 j 가 속한 job class(c)를 기계 m 에서 준비하는데 필요한 셋업시간(setup time),

$\bar{s}_{c'c}$ 작업 j 가 속한 job class(c)가 기계 m 에 투입되기 직전에 그 기계에서 가공을 완료한 임의의 job class(\hat{c})에 대한 평균작업 준비시간(average setup time),

$\hat{s}_{c'c}$ 작업 j 의 예상 셋업시간(expected setup time; EST) 즉, $(1 - P_{cm})s_{c'c} + P_{cm}(s_{c'c} + \bar{s}_{c'c})$,

\hat{p}_j 작업 j 의 예상 가공시간(expected processing time; EPT) 즉, $(1 - P_{cm})p_j + P_{cm}(p_j + \max_{j \in J_c}(p_j))$ 이다.

요약하면 CRR 규칙은 워크센터 앞 대기행렬에 정상적으로 처음 진입한 작업보다 재작업을 위해 재 진입하는 작업에 보다 높은 가중치(weight)를 부여하게 된다. 이 때 해당 작업의 가공시간과 순서의존적인 셋업시간 그리고 납기까지의 상대 여유시간(slack)을 함께 반영할 수 있다.

5. 실험 및 결과분석

5.1. 비교대안 및 실험계획

재작업 확률을 고려한 스케줄링 알고리즘에 관한 기존 연구를 찾아 볼 수 없기 때문에, 본 논문에서는 제안한 알고리즘의 성능평가를 위해 비교 대안 알고리즘으로 기존에 많이 사용되고 있는 작업투입 규칙들을

이용하기로 한다. 가장 대표적인 납기관련 할당 규칙인 EDD(Earliest Due-Dates)와 Lee *et al.*[6]가 제안한 ATCS(Apparent Tardiness Cost with Setups), 그리고 4장에서 소개된 CR 규칙이 그것이다.

실험에 사용된 데이터의 생성 기준은 다음과 같다. 가공 시간과 작업 준비시간은 $U[150,200]$ 의 범위를 갖는 균일분포로부터, 작업투입시점은 0으로부터 평균 최대완료기간(Expected Makespan : EM)의 R배까지인 $[0, R*EM]$ 의 범위를 갖는 균일분포로부터 생성하였다. 여기서 EM은 평균작업 준비시간과 평균가공시간의 합을 작업수 N 과 곱한 후 기계수 M 으로 나누어준 값이며, R은 작업투입시점의 범위모수로서 본 실험에서는 고정값으로 1.0이 사용되었다. 납기는 다음 식(2)에 의해 생성되며, 수식에 사용된 α 값은 $[0, 4]$ 의 범위를 갖는 균일분포로부터 발생된다.

$$d_i = r_i + 2\alpha p_i \quad (2)$$

본 연구의 핵심 요소인 재작업 확률을 실험에 반영하기 위하여 Table 1과 같이 특정 제품군에 속하는 작업타입과 특정 기계간의 재작업 발생확률을 균일분포로부터 생성하였다. Table 1에서 'Lo'는 재작업 가능성이 낮은 상태에서 재작업이 $U[0, 0.01]$ 로부터 발생하는 값(확률 값)에 의해 발생한다는 것을 의미하고, 'Hi'는 재작업 가능성이 높은 상태로 재작업이 $U[0.2, 0.5]$ 에 따라 발생한다는 것을 의미한다. 본 연구의 실험계획 Table 2와 같이 작업수와 제품타입, 그리고 기계수 등 변동될 문제요소의 조합으로 생성된 240개의 문제로 구성되어 있다. 본 연구에서 제시한 알고리즘은 Visual Basic에서 구현하였고, 펜티엄4 2.4 GHz 컴퓨터에서

Table 1. Relationship of rework probabilities between product type and machine

Types	MC	1	2	3	4	5
A		Hi	Lo	Lo	Lo	Lo
B		Lo	Hi	Lo	Lo	Lo
C		Lo	Lo	Hi	Lo	Lo
D		Lo	Lo	Lo	Hi	Lo
E		Lo	Lo	Lo	Lo	Hi
F		Hi	Lo	Lo	Lo	Lo
G		Lo	Hi	Lo	Lo	Lo
H		Lo	Lo	Hi	Lo	Lo
I		Lo	Lo	Lo	Hi	Lo
J		Lo	Lo	Lo	Lo	Hi

Table 2. Experimental design

	Values	Total
No. of Jobs	100, 300, 500	3
No. of Job Types	5,10	2
No. of Machines	3,5	2
Combination		12
Problems per Combination		20
Total Problems		240

실험하였다.

5.2. 실험결과 및 분석

이 절에서는 Table 2에서 제시한 240개의 실험 데이터를 사용하여 본 논문이 제시한 CRR 작업투입 알고리즘의 성능을 평가한다. 이 때 목적함수인 평균지연시간(MT)의 최소화에 대해서 세 가지 대안 알고리즘(EDD, ATCS, CR)과 각각 비교, 분석해 보기로 한다.

해의 질을 가능하는 척도로서 본 실험에서는 3 가지 대안 알고리즘으로 실험하여 얻은 값을 CRR 작업투입 알고리즘으로 얻은 값으로 나누어 준 비교 값(comparison value; CV)을 사용한다. 비교 값 CV는 다음의 식 (3)을 이용하여 구한다.

$$CV = \frac{\text{Average MT of Alternatives}}{\text{Average MT of CRR}} \quad (3)$$

Table 3은 240(12×20)회의 실험결과를 정리한 것이다. 표의 각 셀에 문제와 알고리즘에 따라서 20회씩 반복된 실험의 평균값과 표준편차 값을 표현하였다. Table 3의 결과에서 보듯이 제시한 알고리즘인 CRR 작업투입 알고리즘이 대안 알고리즘인 EDD, ATCS, CR보다 모든 경우에서 좋은 해를 제공하는 것을 볼 수 있다. 특히, 납기 이외의 정보를 전혀 고려하지 못하는 EDD, CR 보다는 월등히 좋은 결과를 도출해내고 있고, 순서의존적인 작업준비시간을 반영하는 ATCS보다도 재작업 확률을 고려하는 본 논문의 알고리즘이 뛰어난 성능을 보여주는 것을 알 수 있다.

6. 결 론

본 논문에서는 작업의 제품타입과 가공되는 기계에 따라 재작업 확률이 달라지는 병렬기계 형태의 반도체 제조공정에서 평균지연시간(MT)을 최소화하는 CRR 작업투입 알고리즘을 제시하였다. 다양한 실험계획에 기반하여 실험을 시행하였고 그 결과 다른 대안알고리

Table 3. Performance comparison between alternatives and proposed algorithm

No of MCs	No of jobs	No of job types	EDD	ATCS	CR	CRR
3	100	5	11.52 ^a (2.78) ^b	4.97 (1.29)	7.79 (3.63)	1.00
		10	10.37 (1.35)	4.20 (1.32)	8.56 (1.33)	1.00
	300	5	13.21 (3.49)	5.21 (2.74)	7.91 (2.87)	1.00
		10	11.22 (1.33)	4.18 (2.77)	7.86 (3.41)	1.00
	500	5	12.96 (3.81)	4.77 (2.95)	7.72 (3.67)	1.00
		10	12.11 (3.79)	5.30 (2.94)	7.43 (3.57)	1.00
5	100	5	13.25 (1.31)	4.98 (1.33)	7.77 (3.06)	1.00
		10	12.56 (4.15)	5.37 (1.3)	7.55 (3.53)	1.00
	300	5	12.03 (4.44)	4.65 (1.32)	7.56 (3.25)	1.00
		10	9.99 (4.12)	3.87 (1.31)	7.10 (3.4)	1.00
	500	5	13.57 (4.26)	4.84 (3.23)	7.09 (1.36)	1.00
		10	12.21 (3.91)	5.33 (1.29)	7.89 (4.26)	1.00

a: Average of total tardiness

b: Standard error of the mean

증인 EDD, ATCS, CR 보다 본 연구에서 제안한 CRR 작업투입 알고리즘이 매우 좋은 결과를 산출하였다. 또한 작업투입 알고리즘의 특성상 우선순위 산출에 소요되는 시간이 매우 빨라서 효율적인 공정제어에 필수적인 실시간적인 의사결정에 효과적으로 적용될 수 있을

것으로 기대한다. 추후에는 CRR 작업투입 알고리즘에 기반한 투입 의사결정의 결과를 제한시간 내에 항상시킬 수 있는 방법에 관한 연구도 필요하다.

참고문헌

1. Flapper, S. D. P. & Teunter, R. H., "Logistic planning of rework with deteriorating work in process," *International Journal of Production Economics*, Vol. 88, pp. 51-59, 2004.
2. Kang Y.H., & Shin H.J, "An adaptive scheduling algorithm for a parallel machine problem with rework processes," *International Journal of Production Research*, Vol. 48, pp. 95-115, 2010.
3. Demirkov, E., & Uzsoy, R., "Decomposition methods for reentrant flow shops with sequence-dependent setup times," *Journal of Scheduling*, Vol. 3, pp. 155-177, 2000.
4. Liu, J.J., & Ping, Y., iOptimal lot-sizing in an imperfect production system with homogeneous reworkable jobs,i *European Journal of Operational Research*, Vol. 91, pp.517-527, 1996.
5. So, K.C., & Tang, C.S., "Optimal operating policy for a bottleneck with random rework," *Management Science*," Vol. 41, pp. 620-636, 1995.
6. Lee, D.H., Ko, H.H., & Kim, S.S., "A study of strategy for planning of rework in semiconductor monitoring burn in test process," *IE interfaces*, Vol. 18, pp. 350-360, 2005.
7. Zargar, A.M., "Effect Of Rework Strategies On Cycle Time," *Computers & Industrial Engineering*, Vol. 29, pp. 239-243, 1995.
8. Sha, D. Y., Hsu, S. Y., Che, Z. H., & Chen, C. H., "A dispatching rule for photolithography scheduling with an online rework strategy," *Computers & Industrial Engineering*, Vol. 50, pp. 233-247, 2006.

접수일: 2011년 2월 25일, 심사일: 2011년 3월 10일,
게재확정일: 2011년 3월 15일