

# 비동기식 선형 파이프라인의 성능 특성 및 이를 이용한 효율적 버퍼할당 알고리즘

\*이정근, \*\*김의석, \*이동익

\*광주과학기술원 정보통신공학과, \*\*광주과학기술원 초고속 광 네트워크 연구센터

## Asynchronous Linear-Pipeline Dynamics and Its Application to Efficient Buffer Allocation Algorithm

\*Jeong-Gun Lee, \*\*Euseok Kim, and \*Dong-Ik Lee

\*Dept. of Info. & Comm., K-JIST, \*\*Ultrafast Fiber-Optic Networks Research Center  
e-mail : eulia@kjist.ac.kr, uskim@kjist.ac.kr and dilee@kjist.ac.kr

### Abstract

This paper presents relationship between the dynamic behavior of an asynchronous linear pipeline (ALP) and the performance of the ALP as buffers are allocated. Then the relationship is used in order to characterize a local optimum situation on the buffer design space of the ALP. Using the characterization we propose an efficient algorithm optimizing buffer allocation on an ALP in order to achieve its average case performance. Without the loss of optimality, our algorithm works in linear time complexity so it achieves fast buffer-configuration optimization.

This paper makes two contributions. First, it describes relationship between the performance characteristics of an ALP and a local optimum on the buffer design space of the ALP. Second, it devises a buffer allocation algorithm finding an optimum solution in linear time complexity.

### I. 서론

비동기식 회로 설계에 있어 데이터-의존형 계산 (data-dependent computation)은 고성능 시스템 설계를 위한 매력적인 장점으로 인식되고 있다 [3,6]. 그러나 이러한 데이터-의존형 계산을 통한 평균-케이스 성능 (average case performance)의 장점은 파이프라인의 스테이지 (stage) 사이의 기아 (starvation) 및 저지 (blocking) 현상으로 인하여 획득하기 어렵다 [2]. 더욱이 파이프라인 구조설계 기법은 명령어나 데이터의 처리에 있어 병렬성을 향상시키는 구조로서 고성능 시스템설계 시에 채택하고 있는 구조이므로 비동기식 파이프라인 구조에서 평균-케이스 성능을 획득하는 것은 중요한 연구 과제이다.

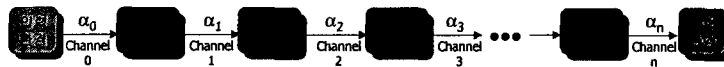
데이터-의존형 계산을 수행하는 비동기 파이프라인의 성능분석 및 기아와 저지현상을 줄이는 버퍼할당과 관련하여 다음과 같은 연구가 진행 되었다.

- "Analyzing and improving the latency and throughput performance of self-timed pipelines and rings" presented by T. E. Williams [1]
- "Performance Evaluation of Asynchronous Logic Pipelines with Data Dependent Processing Delay" presented by D. Kearney [2]
- "Theoretical Limits on the Data Dependent Performance of Asynchronous Circuits" presented by D. Kearney [4]
- "How to Achieve Worst-Case Performance" presented by M. Greenstreet [5]

현재 진행되어온 연구들은 대부분 성능에 대한 근접 분석 (approximate analysis) [2,4], 버퍼와의 연관 관계 유도 [2] 및 데이터-의존형 계산의 한계 분석에 머무르고 있다 [4,5]. 데이터-의존형 계산을 효과적으로 사용하기 위해서 최적의 버퍼할당을 수행하는 자동화된 설계 도구가 요구된다.

본 논문에서는 이를 위하여 비동기식 선형 파이프라인 (asynchronous linear pipeline, ALP)의 버퍼 최적화를 수행하는 알고리즘을 제안하고자 한다. ALP에서는 버퍼 할당에 따라 성능 예측이 어려울 뿐만 아니라 최적 버퍼할당을 찾기 위해서 탐색해야 하는 설계 공간의 크기가 버퍼가 할당될 수 있는 위치의 수와 사용 가능한 버퍼의 수에 따라 빠르게 증가한다. 이는 최적의 버퍼할당 탐색문제를 어렵게 하는 원인이 된다. 본 논문에서는 이를 해결하기 위하여 버퍼할당에 따른 ALP의 동적 동작 및 성능향상 관계를 분석하고, ALP 설계 공간에서 최적의 버퍼 할당 탐색을 어렵게 하는 국부최소 (local optimum)의 발생 상황을 ALP의 성능특성과 관련하여 특성화 (characterization) 하여 탐색 시에 사용함으로써 빠른 탐색을 수행하는 버퍼할당 알고리즘을 구현하고자 한다.

본 논문의 구성으로, 2장에서 ALP에서의 버퍼할당 문제를 구체적으로 정의하고 해결방법에 대한 개관을 설명한다. 3장에서는 ALP의 버퍼 설계공간에서 국부 최소가 발생하는 상황 및 발생하는 원인에 대해서 살펴본다. 4장에서는 ALP 최적화를



$\alpha_i$  = channel i에 할당된 버퍼의 수  
 $BC = [\alpha_0, \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n]$

그림 1 ALP의 구조 및 버퍼배치 (buffer configuration) BC의 구성

위한 버퍼할당 알고리즘의 설계방법을 제안한 후, 5장에서 실험 결과 및 토의로 결론을 맺는다.

## II. 문제정의 및 해결방법에 대한 개관

### • 비용 함수 및 문제 정의

ALP 구조의 버퍼할당 문제와 관련하여 본 논문에서는 설계 비용을 “성능 (performance) × 영역 (area)”으로 정의하였다. 여기서 성능 값은 ALP의 순환시간 (cycle time)으로 정의하며, 영역은 ALP의 영역과 할당된 버퍼의 영역의 합으로 정하였다. 따라서 임의의 버퍼배치 BC에 대하여 비용함수는 다음과 같이 기술된다.

$$\begin{aligned} \text{비용(BC)} &= \text{성능} \times \text{영역} \\ &= \text{순환시간} \times (\text{ALP 영역} + \text{버퍼영역}) \\ &= \text{순환시간} \times (\text{ALP 영역} + \text{버퍼 수} \times \text{단위버퍼영역}) \end{aligned}$$

그림 1은 ALP 구조와 이에 상응하는 버퍼배치 BC를 보여준다. 입력되는 데이터에 따라 가변적인 처리시간을 갖는 n개의 스테이지로 구성된 ALP에는 입/출력 및 스테이지간 데이터가 전달되는 “n+1”개의 채널이 존재하며, 이들 채널에 버퍼가 할당된다. 임의의 채널 i에 할당된 버퍼의 수를  $\alpha_i$ 라고 하면 버퍼배치 BC는 n+1차원 정수벡터로 정의된다. 또한 ALP의 버퍼 설계 공간은 주어진 가용 버퍼의 배치를 통해서 조합할 수 있는 버퍼배치의 집합으로 정의된다. ALP에서 순환시간은 임의의 핸드셰이킹 신호에 대한 변화의 평균시간차이다.

이러한 정의 하에서 버퍼를 할당하는 문제는 “주어진 ALP의 설계 비용을 최소로 하는 버퍼배치 BC를 찾는 문제”로 정의 된다. 그러나 최적의 버퍼배치 BC를 찾는 일은 다음의 두 가지 문제로 인하여 어려워진다.

- ALP의 채널 및 가용 버퍼의 수가 증가함에 따른 버퍼 설계 공간의 폭발적 증가
- 버퍼할당에 따른 ALP 성능예측 어려움

### • 방법론에 대한 개관

ALP 구조에 최적화된 버퍼할당을 하기 위해서 본 논문에서는 시뮬레이션 기반의 반복적인 최적화 (iterative optimization) 작업을 수행한다. 최적의 버퍼배치를 찾기 위하여 버퍼가 할당되지 않은 초기 버퍼 배치를 시작으로 각 채널에 하나의 버퍼를 할당하여 새로운 버퍼배치를 생성한다. 새롭게 생성된 버퍼배치의 집합을 전문 용어로 “이웃집합 (neighbor set)”이라고 부른다. 이웃집합에 속하는 모든 버퍼배치에 대해서 성능 평가를 위한 시뮬레이션을

수행한 후 비용을 계산한다. 이 후 계산된 비용에 근거하여 새롭게 생성된 버퍼배치를 탐색 후보 리스트에 넣고 이를 기반으로 계속 탐색을 할지를 결정한다.

본 논문에서는 버퍼 설계공간에서 국부 최소의 발생 상황을 분석하고 이러한 분석 정보를 통해서 어떠한 버퍼배치를 탐색 후보 리스트에 포함할지 결정한다. 이러한 정보에 근거한 후보결정을 통해 최적성의 희생 없이 많은 버퍼 설계공간을 회피하는 탐색을 수행한다.

## III. ALP 버퍼 설계공간과 국부 최소

거대한 탐색 공간을 갖는 탐색에 있어서 국부 최소의 존재는 최적 해의 탐색을 어렵게 한다. 따라서, 본 장에서 ALP의 버퍼 설계공간에서 국부최소가 어떠한 상황에서 발생하는지 설명 한다.

### • 국부 최소와 추가적 순환시간 감소

ALP의 버퍼 설계공간에서 국부최소가 형성되는 이유를 살펴는 첫 단계로 국부최소 상황을 버퍼 설계 공간에서 구성해보자. 임의의 국부최소 LBC를 버퍼 설계공간에서 가정해 보자. LBC는 국부최소이므로 이웃집합에 속하는 모든 버퍼배치의 비용이 LBC의 비용보다 크다. 국부최소의 정의에 따라 LBC의 비용보다 적은 비용을 갖는 버퍼배치가 LBC로부터 도달 가능하다. 이러한 상황은 LBC로부터 단일 버퍼할당을 통하여 이루지 못한 비용의 감소를 다수의 버퍼 할당을 동시에 고려함으로써 획득 할 수 있다는 것을 의미한다. 비용면에서 “영역”의 값은 버퍼 수에 따라 선형적으로 변화하므로 다수의 버퍼를 동시에 할당하더라도 추가적으로 할당된 버퍼의 수 만큼 선형적으로 증가하여 비용의 면에서 국부최소를 만드는 역할을 하지 못한다. 결국 국부최소의 존재는 순회 시간 (cycle time)의 동적인 변화에 기인함을 알 수 있다.

추가적으로 획득되는 순회시간 감소의 존재 및 그 양을 살펴보기 위해 두개의 버퍼를 ALP의 가능한 채널에 다양하게 할당한 후, “동시에 두개의 버퍼를 할당했을 때 얻은 순회시간”에서 “두개의 개별적인 단일 버퍼의 할당을 통해서 얻은 순회시간의 합”의 차이를 계산하여 동시에 할당이 추가적으로 얻은 순회시간을 계산하였다. 그림 2는 이러한 계산결과를 보여준다. 이 실험은 20개의 스테이지를 갖는 ALP에서 수행하였으며, 각 스테이지의 처리시간은 2와 12사이에서 균일 (uniform) 확률분포를 갖도록 하였다. 두개의 버퍼가 할당되는 채널 위치 (first buffer position과 second buffer position)에 따라 획득되는 순회시간의 차가 다양하게 변화함을 알 수 있다.

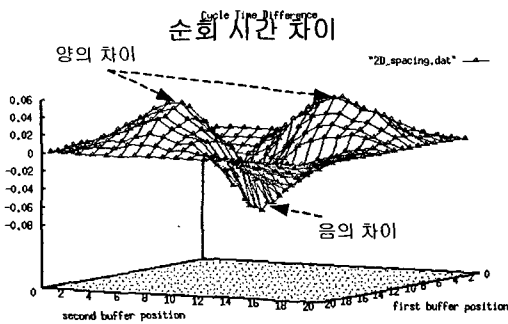


그림 2 동시적 다수 버퍼할당을 통해서 추가적으로 획득되는 순회시간의 차

순회 시간의 차는 할당되는 버퍼간의 상호 위치에 따라 특정한 변화 형태를 보였다. 즉, 할당되는 두개의 버퍼의 위치가 아주 가까울 경우 순회시간의 차가 음의 값이 되고 하나의 버퍼가 특정 위치에 할당 되었을 경우 ALP 을 가장 균형된 처리를 하도록 나머지 버퍼가 할당될 경우 순회시간의 차는 양이 되는 경향을 보이면서 최대 값을 갖았다.

정의 1 (순환시간 차) n 개의 버퍼와 각 버퍼에 대응되는 n 개의 특정 채널이 주어졌을 경우 순환시간 차는 "(동시적 n-버퍼 할당에 의한 순환시간 감소) -  $\sum_n$  (단일 버퍼할당에 의한 순환시간 감소)"로 정의 된다.

정의 2 (추가적 순환시간 감소, ACTR) 양의 순환시간 차를 추가적 순환시간 감소 (additional cycle time reduction, ACTR)로 정의한다.

ACTR 과 국부최소의 관계는 다음과 같이 기술된다.

정리 1 (국소 최소와 ACTR) 버퍼 설계공간에서 LBC 가 국소 최소이면 LBC 에서 다음 조건 만족하며, 역도 성립한다.

- 조건 1) LBC 의 이웃집합에 속하는 버퍼배치의 비용이 LBC 의 비용보다 크다.
- 조건 2) LBC 에서 LBC 보다 적은 비용을 갖도록 만드는 ACTR 을 생성하는 동시적 다수 버퍼할당 존재한다.

정리 1 은 조건 1)은 국부최소 형성을 위한 당연한 조건이다. 중요하게 고려해야 할 사항은 조건 2)에서 언급한 ACTR 이 국부최소와 전역최소의 사이에 탐색에 어려움을 주는 언덕을 설계공간 상에 구성 한다는 사실이다.

• ACTR 의 발생원인

버퍼 설계공간에 국부 최소를 형성하는 중요한 원인이 되는 ACTR 은 버퍼할당에 따른 ALP 의 비선형적 성능특성에 기인한다. 버퍼가 할당된 채널에 스테이지간 상호작용의 완화로 인하여 할당된 버퍼를 기준으로 ALP 구조가 부분적으로 다수의 부분 파이프라인 블록으로 분리 된다 (decoupling). 이때 ALP 의 성능은 분리된 다수의 파이프라인 블록 중

가장 큰 순회시간을 갖는 파이프라인 블록에 의해 크게 결정된다. 이와 같은 비선형적 "max 함수"에 근사한 성능 측정 식과 스테이지 수에 로그적으로 증가하는 순회 시간 성능 특성으로 인하여 ACTR 이 발생하게 된다. 지면의 여유가 없는 관계로 자세한 내용을 실지 못하므로 [7]을 참조하기 바란다.

IV. 버퍼 최적화를 위한 알고리즘

그림 3 에 알고리즘은 ALP 구조에 버퍼를 최적으로 할당한다.

Algorithm Buffer\_Allocation (ALP)

```
{ // n-스테이지 ALP
  buf_Conf[0..n] : integer; // 버퍼배치 데이터 저장
  best_Conf[0..n], LBC[0..n] : integer;
  cost_reduced : boolean; // 비용이 감소에 대한 플래그

  cost_reduced = TRUE;
  while cost_reduced do
  { cost_reduced = FALSE
    for each channel i do
    {
      buf_Conf[i]++; // channel i에 하나의 버퍼 추가
      cost = Perf_Sim(ALP, buf_Conf); // 비용 계산

      // 가장 적은 비용을 가진 버퍼배치를 저장
      if cost < best
      then { best_Conf = buf_Conf;
            best = cost;
            cost_reduced = TRUE;
          }
      buf_Conf[i]--; // 원래의 데이터로 변환
    }
    // 다음 탐색 시작 버퍼배치로 best_Conf 을 사용
    buf_Conf = best_Conf;
  }
  LBC = best_Conf;
  best_Conf = Advanced_Search(LBC);
  return best_Conf;
}
```

그림 3 제안된 버퍼할당 알고리즘

그림 3 에 기술된 알고리즘의 while-loop 에서는 더 이상의 비용 감소가 없을 때까지 for-loop 을 수행 하며 for-loop 에서는 이웃집합을 생성하고 이웃 집합에 속하는 버퍼배치 중 가장 적은 비용을 갖는 버퍼배치를 찾아 다음 while-loop 에서의 시작 버퍼배치로 정한다. 단일 버퍼할당을 통해서 비용이 감소되지 않으면 while-loop 를 나오게 되며 이때의 "best\_Conf(LBC 로 복사됨)" 버퍼배치는 비용을 감소시키는 가능한 모든 단일 버퍼 할당이 수행된 것이며 다수 버퍼 할당에 따른 비용감소가 있는 지 검사하기 위하여 좀 더 발전된 탐색이 "LBC"로부터 시작된다.

"Advanced\_Search" 함수의 입력이 되는 버퍼 배치 LBC 로부터 전역 최소인 버퍼배치로 버퍼할당을 통해 도달 가능해야 한다. 만약 그렇지 못하다면, LBC 로부터 어떠한 탐색기법을 사용하더라도 전역 최소로 갈 수 없게 된다. 제안된 알고리즘의 while-loop 계산을 통해서 찾아낸 버퍼배치 LBC 에서 전역 최소인 버퍼배치로 버퍼할당을 통해 도달 가능함이 다음 정리에 보장된다.

표 1 제안된 알고리즘과 전체설계공간을 찾는 완벽 (exact) 탐색 알고리즘과의 수행시간 성능 비교

파이프라인의 스테이지 수	Exact algorithm			Our algorithm		
	탐색공간 크기	탐색시간	최적 해의 비용	탐색공간 크기	탐색시간	최적 해의 비용
3 (10)	286	38.80	5157.82	19	2.04	5157.82
5 (15)	15504	3454.35	7813.25	61	11.07	7813.25
7 (10)	19448	4878.17	10713.94	71	16.32	10713.94
9 (10)	92378	28616.08	13776.20	91	25.81	13776.20
10 (10)	184756	62490.93	15271.88	101	31.89	15271.88

- ( )속의 수는 가용버퍼의 수를 의미한다.

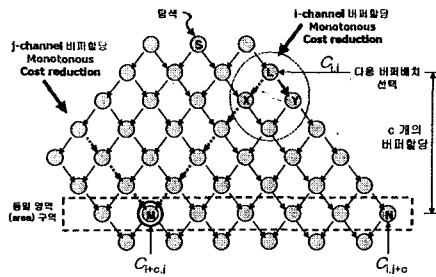


그림 4 LBC로부터 전역최소로 가는 경로

정리 2 (전역 최소와 LBC) 제안된 알고리즘이 계산한 LBC로부터 전역 최소인 버퍼배치로 가는 버퍼할당이 존재한다.

정리 2의 증명 스케치: 그림 4에서 탐색을 통해 버퍼배치 S에 도달했다고 할 때, 전역 최소인 M에 도달 가능하지 않다는 것은 그림 4에서 M으로 향하는 점선 화살표에 의해서 구분되는 위/아래의 버퍼 설계공간 중 탐색이 아래의 버퍼 설계공간에 있음을 의미한다. 임의의 버퍼배치 L에서 Y 방향으로의 선택이 존재했음을 의미한다. 그러나 이러한 선택은 일어날 수 없음이 단일 채널 버퍼할당에 대한 단조적인 성능 특성과 성능 관계식  $u_0 = u - \Delta/c$  [4]의 변형을 통해서 증명하였다 [7].

따라서 초기 버퍼배치로부터 선형탐색을 통하여 LBC로 도달함으로써 많은 설계공간 탐색을 회피할 수 있고, LBC로부터 다수 버퍼할당을 고려한 탐색을 수행함으로써 전역 최소의 버퍼배치로 도달할 수 있다. LBC로부터의 탐색은 그림 3의 알고리즘에 while-loop 내부 계산과 같이 가장 큰 비용 감소를 수행하는 버퍼배치를 선택하는 선형알고리즘 (탐색 정지조건은 [7]을 참조)을 통해서 최적의 해를 찾을 수 있다.

### V. 실험 및 토의

ALP를 성능평가 하기 위하여 사건-중심의 성능 평가도구를 C++언어를 이용하여 개발하였다. 실험 벤치마크로서 3, 5, 7, 9, 10 스테이지의 ALP를 구성한 뒤 개발된 최적화 도구를 통하여 최적의 버퍼할당을 탐색하였다. Table 1은 SUN SPARK 머신에서 수행한 실험결과를 보여준다. 완벽탐색 알고리즘의 "탐색공간 크기"는 최적의 해를 찾기 위해

찾아야 하는 모든 조합의 버퍼배치이다. 실험결과에서 보는 바와 같이 제안된 알고리즘을 통하여 빠른 선형탐색을 수행하였을 뿐 아니라 최적의 버퍼배치를 찾을 수 있었다. 제안된 알고리즘을 통하여 비동기식 파이프 라인 회로의 평균-케이스 성능을 효과적으로 획득할 수 있으리라 기대된다.

### Acknowledgment

본 연구는 한국과학재단의 한일국제공동연구과제 (20006-302-01-2) 및 광주과학기술원 초고속 광네트워크 연구 센터를 통한 한국과학재단 우수 연구센터 지원금에 의한 것입니다.

### 참고문헌

- [1] T. E. Williams. Analyzing and improving the latency and throughput performance of self-timed pipelines and rings. In Proceedings of International Symposium on Circuits and Systems, pages 665-668, vol. 2, May 1992.
- [2] David Kearney. Performance Evaluation of Asynchronous Logic Pipelines with Data Dependant Processing Delays. In Proceedings of the Second Working Conference on Asynchronous Design Methodologies, pages 4-13, London, May. 1995.
- [3] S. M. Nowick, K. Y. Yun et al. Speculative completion for the design of high-performance asynchronous dynamic adders. In Proceedings of the Third International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 210-223, Eindhoven, Apr. 1997.
- [4] David Kearney. Theoretical Limits on the Data Dependent Performance of Asynchronous Circuits. In Proceedings of the Fifth International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 201-207, Barcelona, Spain, Apr. 1999.
- [5] Mark R. Greenstreet and Brian de Alwis. How to Achieve Worst-Case Performance. In Proceedings of the Seventh International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 206-216, Salt Lake City, Utah, Mar. 2001.
- [6] Gianluca Cornetta, Jordi Cortadella. A Multi-Radix Approach to Asynchronous Division. In Proceedings of the Seventh International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 25-34, Salt Lake City, Utah, Mar. 2001.
- [7] Jeong-Gun Lee, Euseok, Kim and Dong-Ik Lee, Efficient Buffer Allocation Algorithms for Achieving Average Case Performance on Asynchronous Pipelines, CSRL-KJIST Internal Report, KJIST, Mar., 2002.