

파이프라인 데이터패스 합성을 위한 스케줄링 알고리즘의 성능평가

오 주 영, 박 도 순 *
* 홍익대학교 컴퓨터 공학과

Estimation of scheduling algorithm's performance for the synthesis of pipelined data path

Ju-Young Oh, Do-Soon Park *
* Dept. of Computer Engineering, Hongik University

요 약

본 논문에서는 자원제약을 목적함수로 하여 파이프라인 실행이 가능하게 하는 데이터패스 합성을 위해 개발되어진 스케줄링 알고리즘들의 실행시간과 실행결과를 대표를 기준으로 정렬한다. 평가의 대상이 되는 알고리즘들은 스케줄을 위해 제안되는 함수의 계산시점, 함수의 역할과 적용방법에 의해 구분되어지는 논문 [1],[2],[3]에 대하여 수행되어지며, 충돌을 발생시키는 파티션 내에 위치하는 모빌리티를 가지는 각각의 연산에 대해 다음 파티션으로의 지연시 충돌수 변이와 각 연산의 모빌리티를 요소로 계산되는 우선 순위 함수를 정의하여 스케줄 순열을 결정하는 논문[1]과 자원 할당 가능성 판단함수를 제안하고 이를 기준으로 배정가능 범위를 축소해 나가며 연산을 스케줄 하는 논문[2]와, 논문[2]의 자원할당 가능성 판단시 부과되는 시간감소를 위해 현재의 스케줄 상황 값들을 정량화 하여 연산이 선택되도록 하여 결과적 실행 시간을 감소시키는 논문[3]에 대하여 벤치마크 성능평가와 알고리즘 실행시간 결과 비교를 수행하고 향후 연구 진행 방향을 제시한다.

1. 서론

상위수준 합성에서의 스케줄링 결과는 설계되는 하드웨어의 수행속도와 면적에 큰 영향을 미치며 다양한 제약사항을 목적함수로 하는 여러 가지 스케줄링 알고리즘들이 개발되어 왔다.

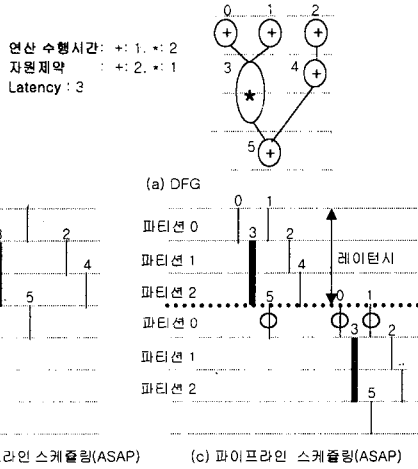
스케줄링 알고리즘은 상위수준 합성, 명령어 수준 병렬실행(ILP)을 위한 소프트웨어 파이프라인, 시스템 병렬디자인(codesign)에서 기본스케줄링블록(BSB)에 대한 자원할당 문제를 해결하기 위하여 하드웨어 면적축적[4]과 하드웨어 실행시간[5] 계산 등에 적용할 수 있다. 디지털 신호처리, 멀티미디어, 데이터 통신시스템 등에서 실행시간 효율을 목적으로 사용되는 파이프라인 데이터패스 합성에 대한 스케줄링 알고리즘 복잡도는 NP-complete[6]문제이며 최소시간 설계법을 위한 리스트 스케줄링, 확률기반 스케줄링, 변환기반 스케줄링 등 많은 휴리스틱 알고리즘들이 개발되었다. 이러한 취지의 일환으로 본 논문에서는 자원 제약 하에서의 파이프라인 스케줄링에 대해서 연구되어진 휴리스틱 스케줄링 알고리즘[1],[2],[3]의 성능과 알고리즘 실행시간을 평가 비교하고 향후 연구 방향을 제시한다. 2절에서 파이프라인 스케줄링 문제를, 3절에서 제안된 스케줄링 알고리즘들의 진행과정을 살펴보고 4절에서 각각의 알고리즘의

성능평가를 5절에서 결론을 기술하였다.

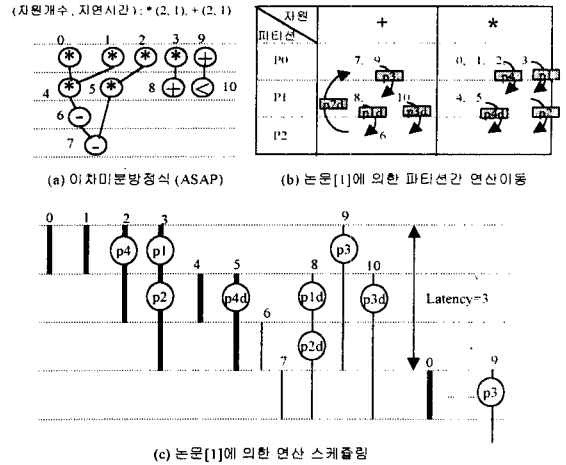
2. 파이프라인 스케줄링 문제

파이프라인 스케줄링에 있어서 가장 중점이 되어야 할 부분은 스케줄 대상 연산에 대한 자원제약과 데이터 입력간격, 이들로 부터 계산되는 레이턴시[7]이며 레이턴시 간격만큼의 파티션에 의해 알고리즘의 대상 탐색 영역이 복잡해지고 스케줄링 결과 또한 DFG 내의 연산수 이상의 제어단계 수로 유도될 수 있다.

파이프라인의 성능($\frac{\text{선형실행시의스테이지수} \times \text{작업수}}{\text{파이프라인스테이지수} + (\text{작업수} - 1) \times \text{레이턴시}}$)은 레이턴시 간격에 의해 크게 좌우되므로 최소 레이턴시 하에서 최적의 해를 유도할 수 있는 낮은 복잡도의 휴리스틱 알고리즘 개발이 필수적이며 최소 레이턴시하에서의 스케줄링 탐색 영역의 변화를 그림[1]에서 찾을 수 있다. 그림[1]의 (a)와 같은 DFG 입력에 대한 비파이프라인 스케줄링(ASAP) 결과는 그림[1]의 (b)와 같지만 파이프라인 스케줄링의 경우는 그림[1]의 (c)에서와 같이 파티션 0에 0번, 1번, 5번 연산이 배정됨으로 인해서 자원제약을 초과하는 충돌이 발생된다. 동일한 DFG의 입력레이턴시 4에 대한 스케줄링 결과는 그림[1]의 (b)와 같게 되지만 파이프라인으로 유입되는 작업의 개수가 많은 경우 레이턴시의 증가에 따른 파이프라인 성능은 선형적으로 감소하게



그림[1] 선형, 파이프라인 스케줄링 비교



그림[2] 우선 순위 함수(논문[1])에 의한 스케줄링

된다. 따라서 최소 레이턴시하에서 최적의 해를 유도할 수 있는 휴리스틱 알고리즘 개발은 필수적이며 이러한 취지에 의해 연구되어진 논문[1],[2],[3]은 비파이프라인 스케줄링 응용과 실질적인 하드웨어 설계 응용을 위하여 체이닝, 멀티사이클링, 연산 모듈의 구조적 파이프라이닝 등을 지원할 수 있도록 설계되어 있다.

3. 스케줄링 알고리즘

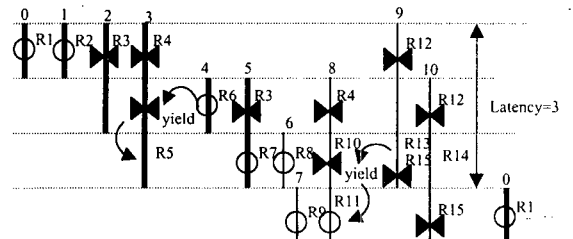
파이프라인 데이터패스 합성을 위해 연구되어진 휴리스틱 방법들(논문[1],[2],[3])은 최적의 결과를 생성하는 효과적인 알고리즘 구현을 위해서 부분 함수를 정의하고 있으며, 이들은 함수의 계산시점과 역할 및 적용방법에 의해 구분되어지며, 대상 연산의 스케줄링을 위하여 정의된 우선 순위 함수의 결과 값에 의해 스케줄링 순열을 단계적으로 결정해 나가는 논문[1]과 자원할당 가능성 판단 함수를 정의하여 각각의 스케줄링 대상 노드들에 대해서 모빌리티 내의 최초·최후 제어단계에 적용하고 불가능한 제어단계를 제거해 나가는 과정에서 연산을 스케줄링하는 논문[2][3]으로 구분할 수 있다.

3.1 논문[1]의 스케줄링 알고리즘

논문[1]은 최초 입력 DFG를 ASAP스케줄 후에 구성되는 파티션의 연산분포를 확인하고 충돌이 발생하는 모든 파티션 내의 모빌리티가 존재하는 각각의 연산들에 대해 지연시 부과되는 충돌개수와 해당 연산의 모빌리티 값을 요소로 하는 우선순위 함수(우선순위함수 = $\sum_{i \in \text{충돌 연산}} (\frac{\text{충돌연산수}_i}{\text{선제자원수}} * \text{모빌리티})$)를 계산하며, 이차미분방정식(그림[2]의 (a))의 DFG 입력에 대해 최소 함수 결과 값에 의한 지연의 순열 번호(pn)와 의존성에 의한 지연(pnd) 번호를 표기하였으며 연산의 스케줄링 결과는 그림[2]의 (c)와 같으며 우선 순위 함수계산은 p4로서 4회 함수 계산에 의해 스케줄링이 완료된다.

3.2 논문[2],[3]의 스케줄링 알고리즘

논문 [2],[3]의 스케줄링 알고리즘은 DFG상의 각 연산노드들에 대해 그 연산의 배정 가능한 제어단계들 중에서 첫 번째와 마지막 제어단계에 대해서 해로의 도달 가능성을 판단하여 스케줄 불가능으로 판단되는 경우 해당 제어단계를 제거하여 배정가능 범위를 축소하여 스케줄 가능성 판단에 있어서 전체적인 탐색 공간을 줄여 휴리스틱의 성공가능성과 시간효율을 높이는데 목적이 있다. 이러한 스케줄링 진행과정에서 모든 연산의 배정범위 축소 후에도 스케줄링이 종료되지 않은 경우 논문[2]에서는 모빌리티가 가장 큰 연산의 ASAP을 선택하여 축소해 나가는 과정으로 스케줄링이 진행되었으며, 논문[3]에서는 현재의 스케줄링 상황(연산의 배정가능 범위, 배정가능범위의 변화량, 연산들의 이동 변화량, 파티션에 연산들의 배정을 균등하게 하는 정보)을 매개변수로 표현하고, 자원충돌 발생 가능성이 적은 연산의 배정 가능 범위를 줄이도록 선택함수를 정의하여 함수의 결과 값에 의해 특정 연산의 제어구간을 줄여 나가도록 한다. 스케줄 진행과정은 최초 제어단계와 마지막 제어단계에 대한 자원할당 가능성 판단을 수행하는 R과 현재의 스케줄링 상황에 의거하여 충돌 가능성이 가장 적은 연산의 배정 가능 범위 축소를 위한 yield를 통해 진행되는 그림[3]과 같다.



그림[3] 배정가능 범위축소(논문[2],[3])에 의한 스케줄링

4. 알고리즘 성능평가

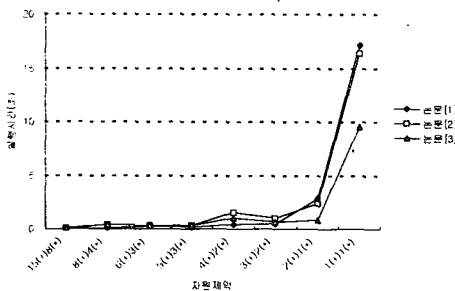
알고리즘 평가를 위한 실험은 인텔 Pentium II MMX 프로세서 환경에서 C++로 수행하였으며 벤치마크는 16포인터 FIR 필터와 5차 엘립틱 웨이브 필터로 하였다. 충돌을 일으키는 파티션 내의 연산에 대해서만 합수계산의 대상으로 하여 스케줄을 진행하는 논문[1]의 실행 시간은 모빌리티를 가지는 모든 연산의 ASAP, ALAP 제어단계에 대한 자원 할당 가능성 판단을 수행하는 논문[2] 보다 우위에 있지만, 충돌 수와 모빌리티에만 의존한 함수의 결과가 항상 최적의 결과가 아님을 그림[4](b)에서 나타내고 있다. 논문[2]는 배정 가능 범위 삭제 과정에서 잔여 모빌리티가 있음에도 더 이상 축약이 되지 않는 경우 모빌리티가 가장 큰 연산의 ASAP을 선택하여 제거하는 방법을 취하였는데, 부적절한 제어구간 삭제로 인하여 자원제약이 심한 경우 최적의 해를 유도할 수 없었다. 이에 대해 논문[3]은 논문[2]의 배정가능 범위 축약의 방법에서 현재의 스케줄 상황 진행표를 근거로 자원할당 검사를 수행하여 제거할 제어단계를 찾아서 최적의 선택이 되게 하였으며 실행 결과는 모든 제약 사항에 대하여 최적의 결과를 보이며 수행 시간 측면에서 자원제약이 심하여 자원충돌이 많아질수록 알고리즘 실행시간(그림[5]) 효율은 좋아지는데 이는 자원할당 가능성 판단시 최선의 연산을 선택하여 제어구간을 할당함으로써 해당 제어구간에 대한 연산의 스케줄링 불가능성 판단이 빠르기 때문이다.

덧셈기 수	15	8	6	5	4	3	2	2	1
곱셈기 수	8	4	3	3	2	2	1	1	1
제어	sehwa	6		6	6	6			
어	Lee[7]	6	6	6	6	6	6	10	16
난	논문[1]	6	6	6	6	6	7	10	15
계	논문[2]	6	6	6	6	6	6	10	16
수	논문[3]	6	6	6	6	6	6	10	15

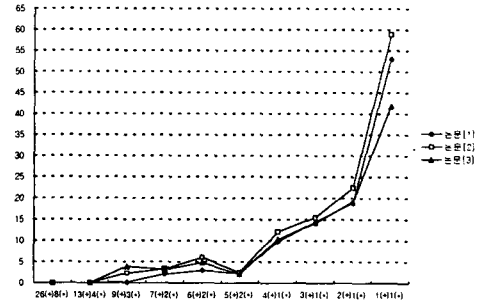
(a) 16포인터 FIR 필터

덧셈기	26	13	9	7	6	5	4	3	2	1	
곱셈기	8	4	3	2	2	2	1	1	1	1	
제어	ILP	17	17	18	19	19	17	20	22	23	33
단	논문[1]	17	17	18	19	19	17	21	22	23	33
계	논문[2]	17	17	18	19	19	17	20	22	23	36
수	논문[3]	17	17	18	19	19	17	20	22	23	33

(b) 5차 엘립틱 웨이브 필터
그림[4] 스케줄링 결과 비교



(a) 16포인터 FIR 필터



(b) 5차 엘립틱 웨이브 필터
그림[5] 알고리즘 실행시간 비교

5. 결론

본 논문에서는 파이프라인 데이터패스 합성을 위해 개발되어진 스케줄링 알고리즘들을 우선 순위 함수의 계산시간과 역할, 적용방법에 의해 구분하고 결과성능과 실행시간으로 비교하여 최적 결과를 위한 최소시간 탐색 영역 알고리즘의 효율을 재고 하였으며 파티션 내의 충돌을 발생시키는 연산들 중에서 지연시킬 최선의 노드를 선택함으로써 재 선택의 시간 비용과 파티션간의 이동을 효율적으로 선택하여 결과성능과 실행시간 측면에서 논문[3]이 최적임을 보였다. 앞으로의 연구 진행 방향은 최적 실행 결과를 위한 최소시간 탐색 영역 알고리즘 개발과 제어합성을 위한 스케줄링 알고리즘 등이 요구되며 현재 설계들의 최소시간 탐색을 위한 알고리즘과 시스템 병렬디자인에 대한 스케줄링 알고리즘 적용에 대한 연구를 진행하고 있다.

참고문헌

- [1] 유동진, 유희진, 박도순, "충돌 수를 이용한 파이프라인 데이터패스 합성 스케줄링 알고리즘", 한국정보처리학회 논문지 제5권 제 11호, 1998
- [2] 윤훈병, 박도순, "배정 가능한 범위 축소에 의한 자원제약 스케줄링 알고리즘", '96 추계학술 발표논문집, 한국정보처리학회, 1996
- [3] Hee-Jin Yoo, Do-Soon Park, "A Scheduling Algorithm for Pipelined data path synthesis with gradual mobility reduction". AP-ASIC, August, 1999 Seoul Korea
- [4] Peter Viogt Knudsen "Fine-Grain Partitioning in Codesign", Master Thesis, Tech. Univ. of Denmark 1995
- [5] Peter Viogt Knudsen, Jan Madsen "FACE : A Dynamic programming algorithm for HW/SW partitioning", 4th Codes/CASHE, 1996
- [6] Giovanni De Micheli, Synthesis and Optimization of digital circuits, McGraw-Hill, 1994
- [7] Y-H Choi, "Synthesis of pipelined datapath". Butterworth-Heinemann Ltd, Computer-Aided Design, 1992