
Retiming을 이용한 Symbolic Model Checking 성능 향상에 관한 연구

강형주*

Improving Symbolic Model Checking Performance Withy Retiming

Hyeong-Ju Kang*

이 논문은 한국기술교육대학교 교육연구진흥비지원 프로그램의 지원에 의하여 수행되었음.

요 약

이 논문에서는 형식 검증(formal verification)의 한 분야인 모델 검증(model checking)에 재타이밍(retiming) 기법을 적용하는 방법에서 대해 연구하였다. 재타이밍은 주어진 회로의 레지스터들을 재배치함으로써, 입출력 동작을 바꾸지 않으면서 전이 관계(transition relation)을 변환할 수 있는 기법이다. 이러한 재타이밍을 이용하면 모델 검증을 더 효율적으로 수행하도록 회로를 바꿀 수 있다. 이 논문에서는, 레지스터의 개수와 전이 관계의 특성을 반영한 cost 함수를 제안하고, 재타이밍으로 얻을 수 있는 회로 구조들을 효율적으로 탐색하는 heuristic annealing 알고리즘을 개발한다. 제안된 방법이 모델 검증의 성능을 향상시킬 수 있음을 실험 결과를 통해 보여주었다.

ABSTRACT

This paper presents an application of retiming to model checking, a branch of formal verification. Retiming can change the transition relation of a circuit without changing its input-output behaviour by relocating its registers. With the retiming, a given circuit can have a different structure more adequate for model checking. This paper proposes a cost function to reflect the number of registers and the characteristic of its transition relation and develops a heuristic annealing algorithm to search efficiently the circuit structures obtained by retiming. Experimental results show that the proposed method can improve the model checking performance.

키워드

모델 검증, 기호적 모델 검증, 형식 검증, 재타이밍

Key word

Model checking, Symbolic model checking, Formal verification, Retiming

* 한국기술교육대학교 (교신저자, hjkang@kut.ac.kr)

접수일자 : 2010. 08. 06

심사완료일자 : 2010. 08. 19

I. 서 론

기능 검증은 디지털 시스템 설계에서 주요 과제이며 설계에 드는 비용의 상당한 부분을 차지하고 있다. 1990 년대에 형식 검증(formal verification)은 기존의 시뮬레이션에 기반한 검증을 보완하는 기법으로 대두되었다 [1]-[3]. 형식 검증의 여러 방법 중 가장 많이 사용되는 것은 모델 검증(model checking)으로써, 주어진 회로에서 어떤 property가 참인지 거짓인지 증명하는 알고리즘이다. 모델 검증에서는 대부분의 시간이 상태 집합의 image나 pre-image를 구하는데 소비된다.

기호적 모델 검증(symbolic model checking)은 상태 집합과 전이 관계(transition relation)를 BDD(Binary Decision Diagram)로 표현함으로써 image와 pre-image를 계산하는 시간을 단축시켰다[2]. 이 방식에서는 상태 집합의 image나 pre-image를 구할 때, BDD를 conjunct시키고 변수들을 quantify하는 방식을 취한다. 전이 관계를 하나의 BDD로 표현할 때 메모리 사용량이 너무 커질 수 있어서, 일반적으로는 각 비트 레지스터들에 대한 비트 전이 관계(bit relation)를 구해서 사용한다. 이러한 비트 전이 관계들을 conjunct시키는 ordering (quantification scheduling)이 image와 pre-image의 계산 시간에 많은 영향을 미치므로, 이에 대해 많이 연구되어 왔다[3]-[7].

기존의 quantification scheduling 알고리즘들은 회로가 주어져 있고 그 구조가 고정되어 있다고 가정해왔다. 그러나 재타이밍(retiming)은 레지스터들을 재배치함으로써 회로의 구조를 동등한 다른 구조로 바꿀 수 있다[8]. 구조를 바꾸게 되면, 레지스터의 개수와 더불어, 각 비트 전이 관계들에 영향을 미치는 변수들도 바뀌고, 전이 관계도 바뀌게 된다. 재타이밍으로 만들어 낼 수 있는 수많은 회로 구조들을 이용하게 되면, image나 pre-image를 계산하는데 더 효율적인 전이 관계를 가지는 회로 구조를 찾을 수도 있고, 따라서 모델 검증을 더 효율적으로 수행할 수 있다.

기존에도 몇몇 연구에서 재타이밍을 모델 검증에 적용했다. D. Stoffel et al.은 [9]에서 순차 검증을 조합 회로 등가 검증(combination logic equivalence checking) 문제로 바꾸는 데 재타이밍을 이용했고, J. Baumgartner et al.은 [10]에서 래치를 옮겨 레지스터로 결합시키기 위해 재타이밍을 이용하였다.

레지스터의 개수를 줄여서 image나 pre-image를 효율적으로 계산하기 위해 재타이밍을 이용하는 방식도 제안되었었다[11]-[13]. 그러나 image나 pre-image 계산의 효율성을 높이기 위해 레지스터의 개수만큼 중요한 것이 전이 관계의 특성이다. 전이 관계의 특성이 좋지 않으면, 레지스터의 개수가 더 적다하더라도, image나 pre-image를 계산하기 위해 더 많은 시간이 걸릴 수도 있다.

이 논문에서는 더 좋은 특성의 전이 관계를 얻기 위해 재타이밍을 사용할 것이다. 전체 알고리즘은 simulated annealing을 조금 수정한 heuristic annealing 알고리즘이다. 재타이밍된 회로의 image와 pre-image 계산 효율성을 반영하기 위해 cost 함수를 제안할 것이다. 이 cost 함수는 변수들의 평균 lifetime과 레지스터 개수의 가중치합이다. 제안된 알고리즘에서는 증명할 property의 참/거짓을 유지하기 위해 입출력 동작을 보존하는 재타이밍 변환만을 사용한다.

이 논문이 다음과 같이 구성되어 있다. 2장에서는 모델 검증 이론과 quantification scheduling, 재타이밍 등에 대해 검토하고, 3장에서 새 알고리즘을 제안할 것이다. 4장에서 실험결과를 제시한 뒤, 5장에서 결론을 맺을 것이다.

II. 연구 배경

기호적 모델 검증에서는 회로가 BDD로 표현된다. 이 논문에서는 회로의 전이 관계를 $T(V, I, V')$ 으로 표기할 것이다. 여기서 V 는 현재 상태 변수들의 집합이고, I 는 입력 변수들의 집합이며, V' 은 다음 상태(next state) 변수들의 집합이다. 현재 상태의 집합은 현재 상태 변수 집합인 V 로 표현되며, 다음 상태의 집합은 다음 상태 변수 집합인 V' 으로 표현된다. 상태 집합 $Sa(V)$ 의 image $Sa'(V')$ 은 다음과 같이 계산할 수 있다.

$$S'_a(V') = \exists V, I. (S_a(V) \cdot T(V, I, V')) \quad (1)$$

모델 검증 알고리즘은 상태 집합간의 집합 연산들과 위와 같은 image나 pre-image 계산들로 이루어져 있다. 이 논문에서는 image 계산을 중심으로 논할 것이나, 제안된 알고리즘은 pre-image 계산에도 적용될 수 있다.

2.1 Quantification Scheduling

하나의 BDD로 전체 전이 관계 $T(V,I,V')$ 를 모두 표현하는 것은 메모리를 너무 많이 소모한다. 그래서 다음과 같이 암묵적으로 conjunct되어 있는 일련의 비트 전이 관계들로 표현한다.

$$T(V,I,V') = \bigwedge_{0 \leq i < n} T_i(V,I,v'_i) \quad (2)$$

비트 전이 관계들이 식 (1)의 image 계산에 사용되면, 일부 변수들은 최종 결과가 나오기 전에 quantify될 수 있다. 각 비트 전이 관계 $T_i(V,I,v'_i)$ 에 대해 D_i 를, $T_i(V,I,v'_i)$ 에 들어 있는 $I \cup V'$ 의 변수들이라고 가정하고, E_i 를 다음과 같이 정의하자.

$$E_i = D_i - \bigcup_{k=i+1}^{n-1} D_k \quad (3)$$

그러면 식(1)은 식(2)와 (3)을 이용해서 다음과 같은 방식으로 계산될 수 있다.

$$\begin{aligned} P_0 &= S_a(V) \\ P_i &= \exists E_{i-1}. (P_{i-1} \cdot T_{i-1}(V,I,v'_{i-1})) \\ &\quad \text{for } 0 < i \leq n \\ S'_a(V) &= P_n \end{aligned} \quad (4)$$

Conjunct 연산과 quantify 연산의 연속인 위 계산의 과정에서 만들어지는 BDD의 가장 큰 크기는, $T_i(V,I,v'_i)$ 들이 conjunct되는 순서와 밀접한 관련이 있다. 순서가 다르면 E_i 가 달라지고, 따라서 계산중에 quantify되지 않고 남아 있는 변수의 개수도 달라진다. BDD의 크기가 변수의 개수와 관계가 있으므로, $T_i(V,I,v'_i)$ 들이 conjunct되는 순서에 의해 BDD의 크기가 영향을 받는다.

최적의 순서를 찾는 것을 quantification scheduling 문제라고 부르며, 이에 대해 많이 연구되어 왔다. 그 중, 가장 널리 사용되고 있는 방식은 다음과 같은 의존 행렬 (dependence matrix)을 이용하는 방식이다. 다음의 정의들은 [3]을 참고하였다.

정의 1. 변수 x_1, \dots, x_n 로 이루어진 순서가 있는 함수들 f_1, \dots, f_m 의 의존 행렬은, 행의 개수가 m 개이고 열의 개수가 n 개인 행렬 D 로써, 함수 f_i 가 변수 x_j 를 가지고 있을 때 D 의 원소 $d_{ij}=1$ 이 되고, 그렇지 않으면 $d_{ij}=0$ 이다.

I.-H. Moon et al.은 [3]에서 행렬의 행과 열을 바꾸면서, 다음에서 정의할 lifetime을 최소로 하는 ordering을 찾는다.

정의 2. 의존 행렬 D 의 total lifetime은 다음과 같이 정의된다.

$$\lambda = \frac{\sum_{1 \leq j \leq n} (m - l_j + 1)}{n \cdot m} \quad (5)$$

여기서 l_j 는 열 j 에서 $d_{ij}=1$ 인 최소 i 이다. 비슷하게 active lifetime은 다음과 같이 정의된다.

$$\alpha = \frac{\sum_{1 \leq j \leq n} (h_j - l_j + 1)}{n \cdot m} \quad (6)$$

여기서 h_j 는 열 j 에서 $d_{ij}=1$ 인 최대 i 이다.

2.2 재타이밍

재타이밍은 레지스터들을 재배치함으로써 회로의 구조를 바꾸는 것으로써, 순차회로의 최적화에 많이 사용된다. 처음 제안되었을 때에는 critical path의 지연 시간이나 레지스터의 개수를 줄이는 것이 목적이었다[8]. 그러나 그 이후로 이용되는 곳이 많아져서, 재타이밍 후 재합성(retiming and resynthesis)이나 주변 재타이밍(peripheral retiming)과 같은 목적으로도 이용된다.

레지스터들을 재배치한 후에 negative 레지스터가 없으면 그 재타이밍은 적법하다고(legal) 부른다. 적법한 재타이밍을 통해 얻은 회로는 재타이밍 이전의 회로와, 입출력 동작의 면에서 보았을 때, 동일하다고 알려져 있다. 이 논문에서는 입출력 동작을 유지하면서 모델 검증의 성능을 향상시키기 위해 적법한 재타이밍을 사용할 것이다.

III. 기호적 모델 검증을 위한 재타이밍

이 장에서는 모델 검증에 더 효율적인 재타이밍된 회로 구조를 찾는 알고리즘을 제안할 것이다. 재타이밍은 각 비트 전이 관계에 포함된 변수들을 바꾸어서, *image* 계산의 복잡도를 높이거나 낮출 수 있고, 그러므로 모델 검증의 성능도 바꿀 수 있다. 이 논문에서 제안할 알고리즘은 주어진 회로에 여러 가지의 재타이밍을 시도하면서, 3.1절에서 제안할 *cost* 함수를 통해 그 재타이밍된 회로의 *image* 계산 복잡도를 예측하며, *heuristic annealing* 과정을 통해 더 나은 재타이밍된 구조를 찾는다.

제안할 알고리즘은 일종의 전처리 과정의 역할을 한다. 주어진 회로를 재타이밍을 통해 최적화한 뒤, 검증 알고리즘이 재타이밍된 회로로 *property*를 증명하는 것이다. 이 논문에서는 이 두 단계를 각각 재타이밍 단계와 검증 단계로 부를 것이다.

3.1 Cost 함수

모델 검증에 적합한 회로를 찾기 위해서는 재타이밍된 회로가 얼마나 적합한지를 평가할 수 있는 *cost* 함수가 필요하다. 일반적인 재타이밍 알고리즘에서는 주로 레지스터의 개수나 *critical path*의 지연 시간이 *cost* 함수로 사용되나, 이러한 *cost* 함수들은 *image* 계산의 효율성을 나타내기에 적합하지 않다. 제안하는 알고리즘의 *cost* 함수는 재타이밍된 회로의 전이 관계의 특성을 나타낼 수 있어야 한다.

Image 계산의 복잡도에 영향을 미치는 것 중의 하나는 레지스터의 개수이다. 레지스터의 개수가 적으면, 전이 관계가 더 적은 개수의 비트 전이 관계로 구성되고 *image*가 더 적은 수의 상태로 표현되므로, 더 선호될 것이다.

레지스터의 개수만큼 영향을 많이 미치는 것이 각 비트 전이 관계의 특성과 비트 전이 관계들 사이의 관계이다. 식 (4)를 계산하는 복잡도를 줄이기 위해서는, 각 비트 전이 관계들이 더 적은 개수의 변수들로 이루어지고 더 간단한 형태의 *BDD*로 표현되고 있어야 한다. 그리고, 식 (4)의 각 단계에서 더 많은 변수들을 *quantify*하기 위해서는, 각 비트 전이 관계들이 더 적은 수의 변수들을 공유하는 것이 좋을 것이다. I.-H. Moon

et al.은 [3]에서 이러한 특성을 *lifetime*으로 계량화하였다.

위에서 언급한 두 가지의 요인은 서로 모순된 관계에 있다. 예를 들어 레지스터가 2입력 게이트를 가로질러서 움직인다고 가정해 보자. 레지스터가 게이트의 입력 A, B에 있다가 전진해서 출력 C로 움직였다면 레지스터의 개수는 줄어든다. 그러나 C에 대한 비트 전이 관계는, A의 비트 전이 관계에 들어 있던 변수들과 B의 비트 전이 관계에 들어 있던 변수들의 합집합인 변수들로 구성된다. 이렇게 되면 보통 C의 비트 전이 관계는 더 복잡해지고 의존 행렬의 *lifetime*도 커진다.

이런 모순관계인 두 가지의 요인을 동시에 고려하기 위해, 본 논문에서는 두 요인의 가중치 합을 *cost* 함수로 사용하였다. 그 식은 다음과 같으며, 여기서 #은 개수를 의미한다.

$$\text{cost} = w_1 \frac{\text{lifetime after retiming}}{\text{initial lifetime}} + w_2 \frac{\# \text{ of registers after retiming}}{\text{initial \# of registers}} \quad (7)$$

3.2 Property 보존

모델 검증에서 재타이밍을 이용하기 위해서는 *property*의 참/거짓이 재타이밍된 회로에서도 유지되어야 한다. 적법한 재타이밍은 입출력 동작을 유지하므로, *property*가 회로 출력의 함수라면 그 참/거짓은 유지될 것이다. 만일 *property*가 회로의 내부 신호로 표시된다면, 그 내부 신호들을 회로의 출력으로 나가는 것처럼 임시로 바꾼 뒤 재타이밍을 적용함으로써, *property*의 참/거짓을 유지시킬 수 있다.

IV. 전체 알고리즘

이 장에서는 전체 알고리즘에 대해 설명할 것이다. 하나의 회로에서 재타이밍을 통해 만들어 낼 수 있는 회로는 아주 많다. 이러한 다양한 회로들을 효율적으로 탐색하기 위해, 이 논문에서는 *heuristic simulated annealing*에 기반하여 알고리즘을 구성하였다. 재타이밍을 이용하는 기존의 알고리즘들은 주로 *linear programming*이나

longest path search 알고리즘에 기반하고 있다. 이러한 알고리즘들은 cost 함수가 레지스터들의 위치로부터 쉽게 계산할 수 있을 때 유용하다.

그러나, 이 논문에서 제안하는 알고리즘의 cost 함수는 쉽게 계산하기 어려우므로, 기존의 방법은 적용하기 어렵다. 제안하는 알고리즘에서는, 재타이밍된 여러 회로들, 즉 여러 solution들을 탐색하기 쉽고 최종적으로 찾은 solution과 시간 사이의 trade-off가 용이한 simulated annealing을 조금 변경하여 사용하였다.

알고리즘이 시작되면, 우선 주어진 회로의 의존 행렬을 추출한다. 그 다음에 재타이밍 과정을 진행한다. 제안하는 알고리즘에서 하나의 solution은 재타이밍된 회로와 그 회로의 의존 행렬로 구성된다. 현재의 solution에서 레지스터를 옮기면 새로운 solution이 나온다. 이 새로운 solution의 cost가 이전의 solution보다 작으면, 새로운 solution이 현재의 solution이 된다. 그러나, 새로운 cost가 이전의 cost보다 크면, 새로운 solution이 현재 solution이 될지는 확률에 따른다. 이 확률은 이전 cost와 새 cost의 차이가 작을수록 커지며, 알고리즘을 반복할수록 점점 작아지도록 하였다. 일정한 횟수의 반복을 한 뒤에, 그 때까지 찾은 가장 낮은 cost값을 검사한다. 이전에 검사했을 때에 비해 cost값의 개선이 이루어지지 않거나 너무 오랜 시간이 지나면 알고리즘을 종료한다.

V. 실험 결과

ISCAS89 벤치마크 회로들 중 일부를 이용하여 제안한 알고리즘의 성능을 비교하였다. 표 1은 사용한 벤치마크 회로들의 특성들이다. 첫 번째 열이 회로들의 이름이고, 두 번째 열에서 네 번째 열까지는, 입력 개수, 출력 개수, 레지스터 개수이다. 레지스터들을 처음에 0으로 초기화시키는 리셋 회로를 벤치마크 회로들에 추가하였다.

제안한 알고리즘은 C로 구현하였으며 GCC로 컴파일하였다. VIS[15]를 이용하여 원 회로와 재타이밍된 회로의 상태 도달성 분석을 실행하였다. 실험은 3GHz CPU를 탑재한 리눅스 머신에서 시행하였다. 상태 도달성 분석은 초기 상태(initial state)로부터 시작해서 도달이 가

능한 모든 상태들을 구하는 과정으로써, image 계산 능력을 비교하기에 편리하기 때문에 모델 검증 알고리즘들의 성능을 비교하기 위해 많이 이용된다.

식 (7)의 cost 함수에서 가중치는 $w_1=3/4$, $w_2=1/4$ 를 이용하였다.

표 1. 실험 대상 회로
Table. 1 Circuits for Experiments

회로 이름	입력 개수	출력 개수	레지스터 개수
s1269	18	10	38
s1512	29	21	57
prolog	36	73	137
s4863	49	16	104

표 2. 실험 결과
Table. 2 Experimental Results

회로 이름	원 회로		재타이밍된 회로	
	시간 (s)	메모리 (MB)	시간 (s)	메모리 (MB)
s1269	1443	137	535	98
s1512	214	71	191	69
prolog	3710	113	1373	122
s4863	7966	130	496	94

표 2에서 원 회로와 제안된 알고리즘으로 재타이밍된 회로의 상태 도달성 분석을 비교하였다. 첫 번째 열은 회로들의 이름이고, 그 다음 2개의 열이 원 회로의 상태 도달성 분석에 소요되는 시간과 메모리 사용량이고, 다음 두 개의 열이 제안된 알고리즘으로 재타이밍된 회로 상태 도달성 분석하는데 소요되는 시간과 메모리 사용량이다. 표 2의 결과를 비교하면, 제안된 알고리즘으로 재타이밍할 경우, 상태 도달성 분석에 소요되는 시간과 메모리가 감소함을 알 수 있다. 특히, s4863회로의 경우에는 시간이 1/10 이하로 감소하였다.

VI. 결론

이 논문에서는 모델 검증에 재타이밍을 이용하는 기법에 대해서 연구하였다. 모델 검증에 대한 기존의 연구에서는 주어진 회로를 표현하는 방법과 image나 pre-image를 계산하는 방법에 대해 주로 연구하여왔고, 재타이밍에 대한 기존 연구에서는 주로 면적이나 critical path의 지연 시간을 줄이는 것에 대해 연구하여왔다. 이 논문에서는, 이 두 영역을 연결해서 재타이밍을 통해 모델 검증에 알맞은 전이 관계를 가지도록 회로를 변환하는 방법에 대해 제안하였다. 이를 위해 image 계산의 어려움을 나타내는 cost 함수와, 재타이밍으로 만들어지는 여러 회로들을 효율적으로 탐색할 수 있는 heuristic annealing 알고리즘을 제안하였다. 실험 결과를 통해, 제안한 알고리즘이 모델 검증에 더 알맞은 회로 구조를 찾을 수 있음을 보였다.

참고문헌

- [1] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*, MIT Press, MA: Cambridge, 1999.
- [2] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and L. J. Hwang, "Symbolic model checking: 1020 states and beyond," *Proc. Symp. Logic in Comput. Sci.*, pp. 428-439, 1990.
- [3] I.-H. Moon, G. D. Hachtel, and F. Somenzi, "Border-block triangular form and conjunction schedule in image computation," *Proc. Int. Conf. Formal Methods in Computer Aided Design of Electron. Circuits.*, pp. 73-90, 2000.
- [4] I.-H. Moon, J. H. Kukula, K. Ravi, and F. Somenzi, "To split or to conjoin: the question in image computation," *Proc. ACM/IEEE Design Automation Conf.*, pp. 23-28, 2000.
- [5] R. K. Ranjan, A. Aziz, R. K. Brayton, B. F. Plessier, and C. Pixley, "Efficient BDD algorithms for FSM synthesis and verification," *Proc. Int. Workshop on Logic Synthesis*, 1995.
- [6] D. Geist and I. Beer, "Efficient model checking by automated ordering of transition relation partitions," *Proc. Int. Conf. Computer Aided Verification*, pp. 299-310, 1994.
- [7] R. Hojati, S. C. Krishnan, and R. K. Brayton, "Early quantification and partitioned transition relations," *Proc. Int. Conf. Computer Design*, pp. 12-19, 1996.
- [8] C. Leiserson and J. Saxe, "Retiming synchronous circuitry," *Algorithmica*, pp. 5-35, 1991.
- [9] D. Stoffel and W. Kunz, "Record & play: a structural fixed point iteration for sequential circuit verification," *Proc. Int. Conf. Computer Aided Design*, pp. 394-399, 1997.
- [10] J. Baumgartner, T. Heyman, V. Singhal, and A. Aziz, "Model checking the IBM gigahertz processor: an abstraction algorithm for high-performance netlists," *Proc. Int. Conf. Computer Aided Verification*, Lecture Notes in Computer Science, vol. 1633, pp. 73-83, 1999.
- [11] G. Cabodi, S. Quer, and F. Somenzi, "Optimizing sequential verification by retiming transformations," *Proc. ACM/IEEE Design Automation Conf.*, pp. 601-606, 2000.
- [12] A. Kuehlmann and J. Baumgartner, "Transformation-based verification using generalized retiming," *Proc. Int. Conf. Computer Aided Verification*, Lecture Notes in Computer Science, vol. 2102, pp. 104-117, 2001.
- [13] A. Gupta, P. Ashar, and S. Malik, "Exploiting retiming in a guided simulation based validation methodology," *Proc. Correct Hardware Design and Verification Methods*, Lecture Notes in Computer Science, vol. 1703, pp. 350-353, 1999.
- [14] V. Singhal, S. Malik, and R. K. Brayton, "The case for retiming with explicit reset circuitry," *Proc. Int. Conf. Computer Aided Design*, pp. 618-625, 1996.
- [15] The VIS Group: VIS 2.0. [Online]. Available: <http://vlsi.colorado.edu/~vis/>.

저자소개



강형주(Hyeong-Ju Kang)

1998년 한국과학기술원
전기및전자공학과 학사
2000년 한국과학기술원
전기및전자공학과 석사

2005년 한국과학기술원 전자전산학과 박사
2005년~2006년 (주)매그나칩반도체 선임연구원
2006년~2010년 (주)지씨티리씨치 선임연구원
2010년~현재 한국기술교육대학교 컴퓨터공학부
전임강사

※관심분야: VLSI설계 및 CAD, 마이크로프로세서
설계, 통신 모델 설계