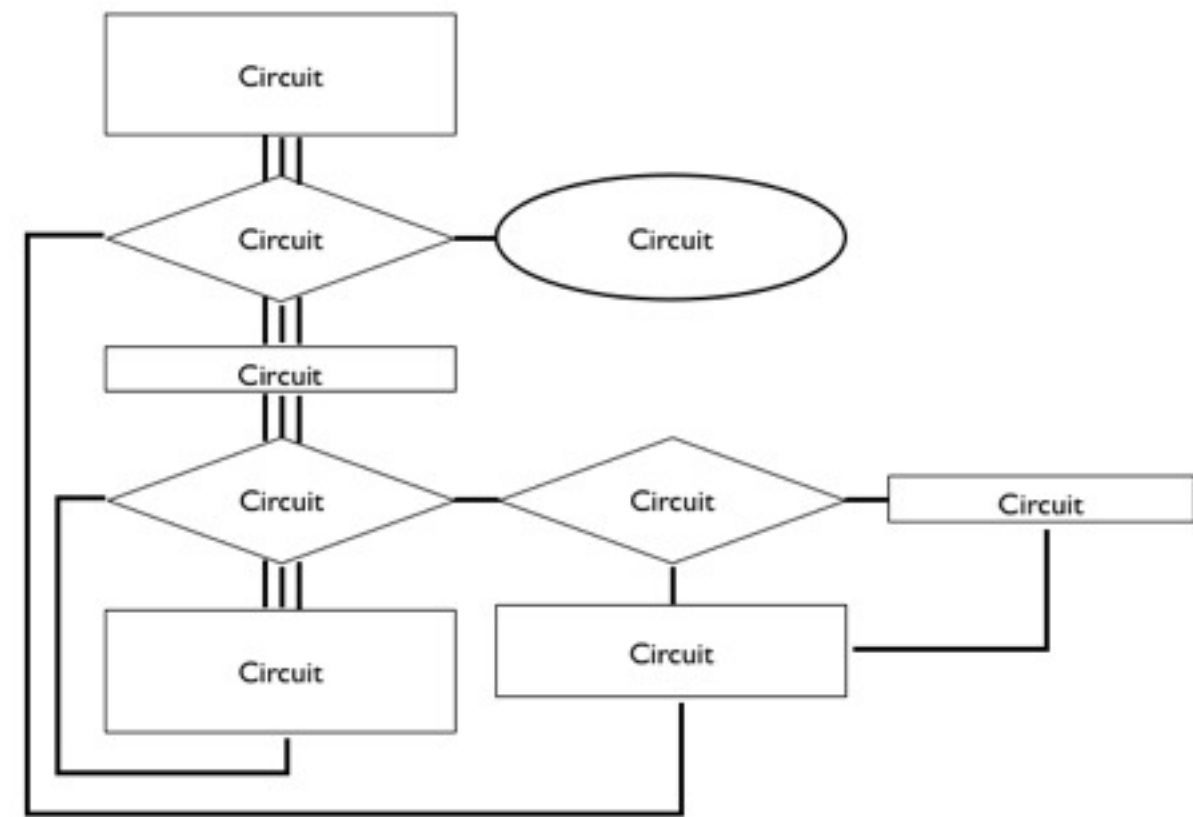
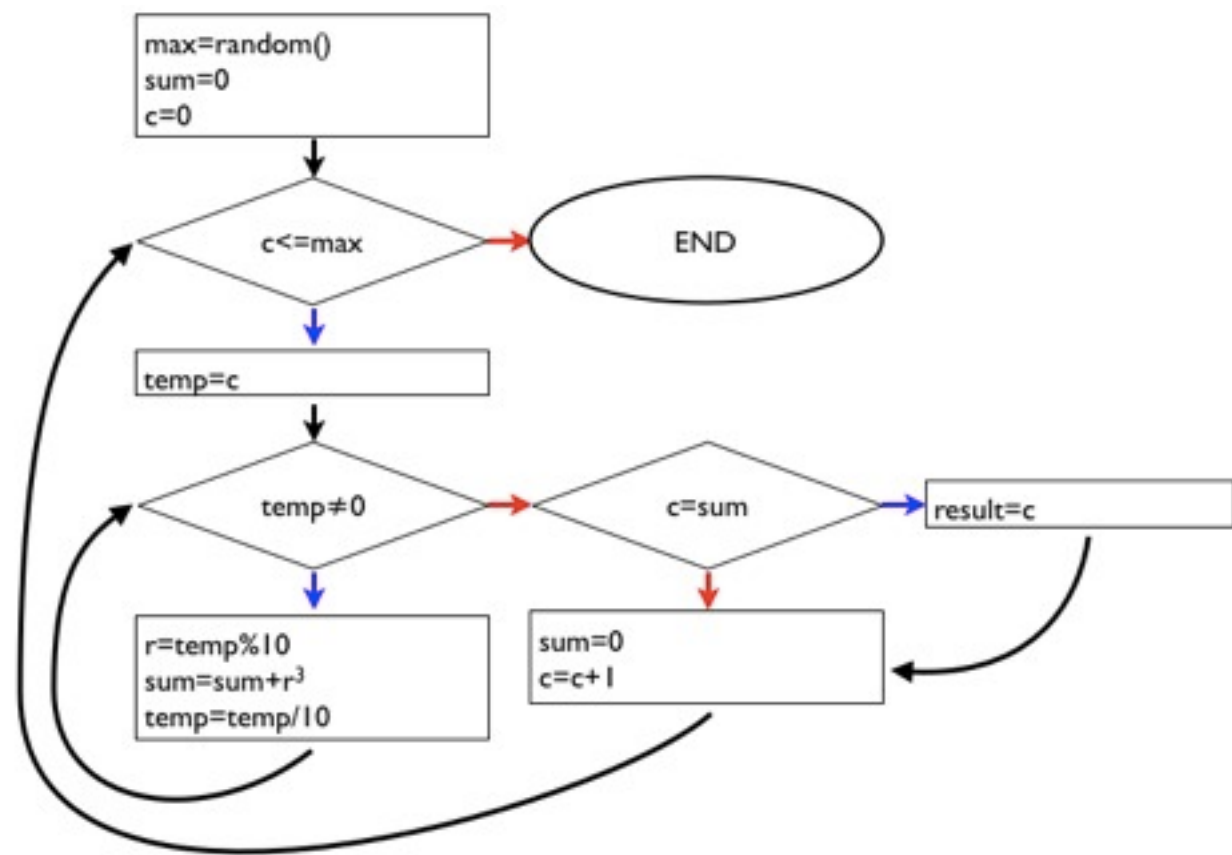


프로그램의 정적 분석을 디지털 회로로 표현하기

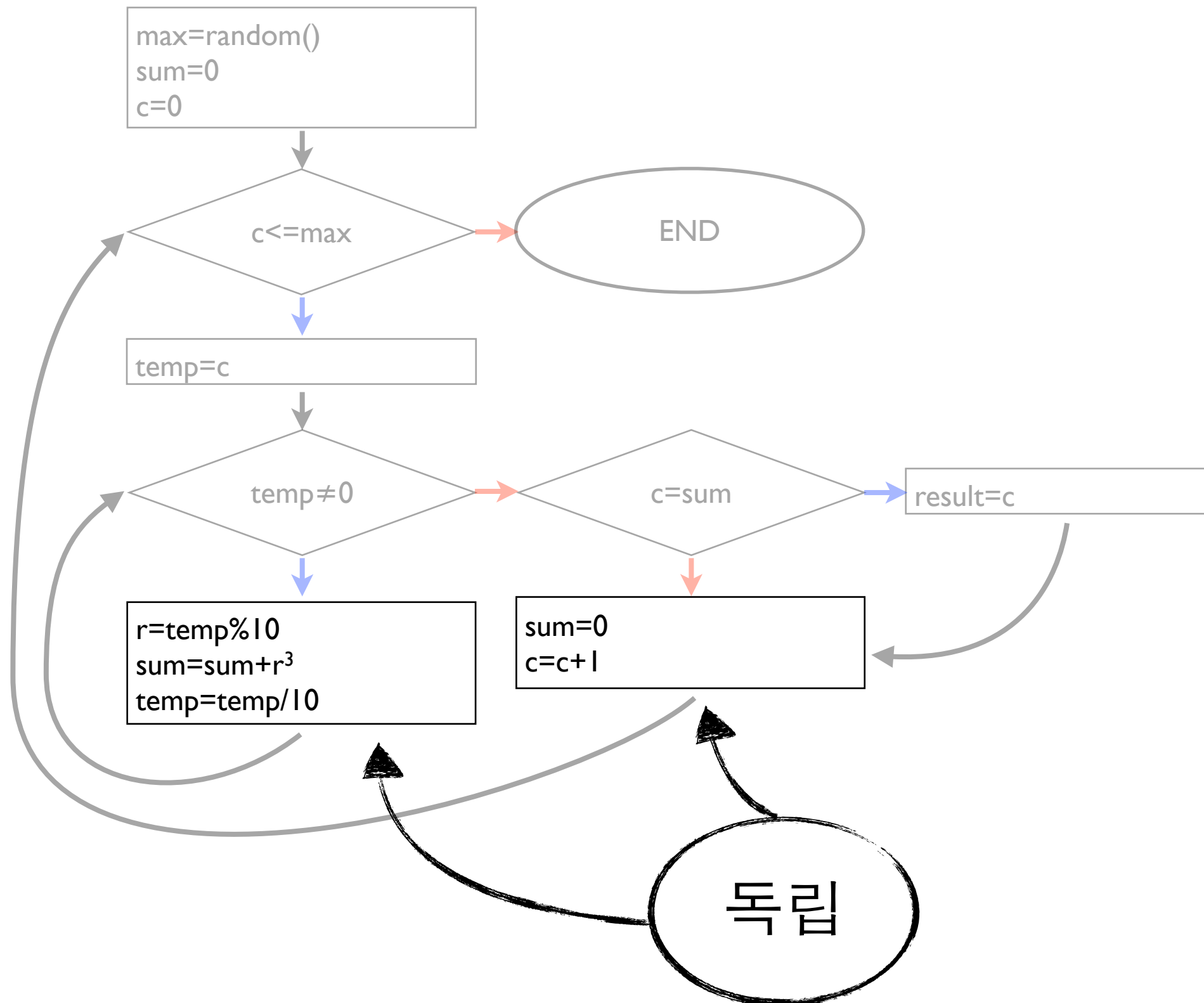
서울대학교 프로그래밍연구실
윤용호

yhyoon@ropas.snu.ac.kr

동기 1



동기 2



정적 분석 + 디지털 회로

- 모든 계산을 동시에, 빠른 수렴
- 새로운 시도, 재미

디지털 회로 설계

- HDL(~~Hard Dirty Language~~ Hardware Description Language)
 - Verilog, VHDL, ...
- 시뮬레이션 용 바이너리
- 실제 회로 합성(Synthesis)

디지털 회로 설계

- HDL(~~Hard Dirty Language~~ Hardware Description Language)
 - Verilog, VHDL, ...
- 시뮬레이션 용 바이너리
- 실제 회로 합성(Synthesis)



합정!

합정

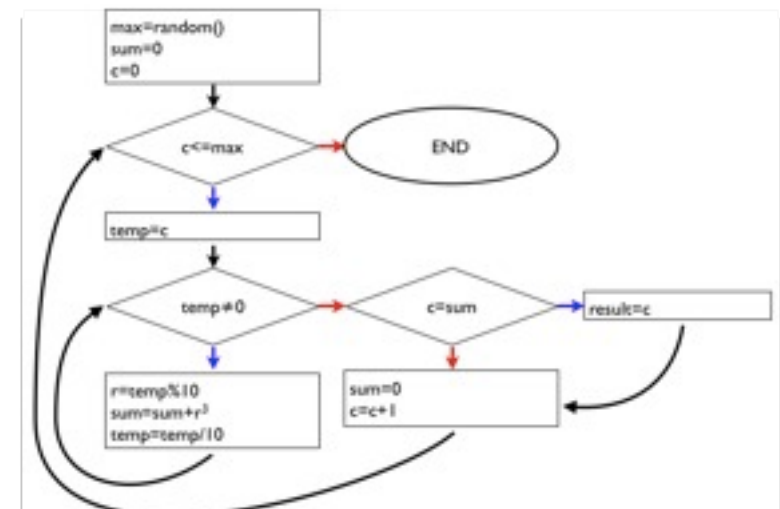
- 일반적으로 합성은 굉장히 오래 걸림
 - 게이트 할당, 도선 배치, 클럭 어긋남 관리, ...
- 회로 합성 시간 >>>> 분석 시간?

합성 피하기 고민

- 합성 해두고 계속 사용
 - 대규모 프로젝트의 안정화된 코드
 - 라이브러리 코드
- 합성 시간 줄이기
 - 미리 만들어둔 블록 최대한 활용
 - FPGA 모방

첫 걸음, 손으로

- 간단한 프로그램을 회로로 옮겨봄
- Verilog 코드 30KB
- 18사이클에 수렴



/TB_TOP/top/block3/state_to_var/r	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block3/state_to_var/result	1000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block4/state_to_var/max	00000001400000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block4/state_to_var/sum	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block4/state_to_var/c	1000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block4/state_to_var/temp	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block4/state_to_var/r	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block4/state_to_var/result	1000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block5/state_to_var/max	00000001400000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block5/state_to_var/sum	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block5/state_to_var/c	1000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block5/state_to_var/temp	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block5/state_to_var/r	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block5/state_to_var/result	1000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block6/state_to_var/max	00000001400000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block6/state_to_var/sum	3000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block6/state_to_var/c	1000000000000000...	4000000000000000	1000000000000000	1000000000000000
/TB_TOP/top/block6/state_to_var/temp	3000000000000000...	4000000000000000	1000000000000000	1000000000000000

앞으로

- Sparse 분석을 하는 회로
 - 레지스터 수는 메모리 크기보다도 큰 제약
 - 회로가 스스로 Sparse 분석을 구성하는 것이 가능한가?
- C → Verilog 코드 자동 생성기
 - 실제 회로로 합성(Synthesis) 가능한 코드

고맙습니다

- 포스터 발표도 합니다 -