

VDSM 공정에서 적용되는

RTL-to-GDS Methodology 검토 및 적용

권오철, 정길임, 김주선, 배점만

삼성전자 중앙연구소 NS Lab.

전화:031-200-3343 / 이동전화:017-203-5753

Review on RTL-GDS Methodology for VDSM Process

O Chul Kwon, Gil Im Jung, Joo Sun Kim, Jum Han Bae

NS Lab., Corporate R/D Center, Samsung Electronics

{ ochulkwon, snjskim, jhbae }@samsung.com

jkl@asic.sec.samsung.co.kr

Abstract

We have been aware for some time, that it is becoming harder to develop ASIC only, using the vendor wire model for the current top-down/bottom-up process. Because VDSM has a much bigger wired delay than cell delay, it is also difficult to reduce development time, as well as time-to-market, while developing several million gate ASIC's. The same is true for high frequency ASIC's with VDSM (which have larger wire delay versus cell delay). Therefore, a solution called "RTS-GDS", using physical constraints for SOC with timing met, is being actively discussed. This paper suggests a methodology for SOC development by utilizing a top down flow via CWLM along with discussing potential problems. This paper also provides a design flow, including physical synthesis, DFT, floor plan and CWLM, all of which are relevant to proper SOC development.

I. 서론

SOC 구현을 위한 ASIC 개발의 최대 issue는 상위 level에서의 system design과 RTL-to-GDS 구현임을 알고 있다. 본 논문에서는 RTL-to-GDS의 single pass design에 관한 methodology와 이때 야기되는 문제점 review에 중점을 두었으며 현재의 tool 환경으로 구현 가능한 점과 어떤 flow가 빠른 time-to-market에 대응할 수 있는지를 논의하고자 한다. 본 논문에서는 현재 Mega급

ASIC 개발에 적용되는 wire delay를 이용한 POST iteration 기법인 Links To Layout Flow에 대한 내용을 살펴보고 3M gate급 이상의 SOC 구현을 위한 methodology 적용 사례와 결과를 통하여 문제점과 향후 flow 방향에 대해서 알아보고자 한다. 또한 최근 논의되고 있는 physical synthesis 개념을 도입하고 관련하여 AVANTI사의 Jupiter tool의 demo를 통하여 SOC Chip Implementation Methodology를 소개하고자 한다.

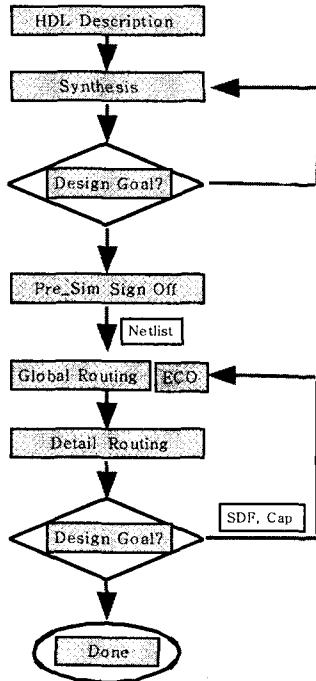
1-1. Traditional Design Flow

<그림 1>은 vendor에서 제공하는 wire load model을 이용하여 synthesis 후 Pre simulation sign-off된 netlist를 이용하여 FloorPlan을 수행하고 global routing 및 detail routing을 수행 후 SDF(Standard Delay Format), cap file을 back annotation하여 STA(Static Timing Analysis)를 수행하는 flow를 보여 주고 있다. 그러나 이 methodology는 0.5 μ m 이상의 design rule이 적용된 경우에는 cell delay만을 고려한 post simulation에 문제가 없으나 VDSM 하에서는 cell delay보다는 60% 이상의 wire delay에 의한 timing 지연 요소가 발생하므로 많은 post iteration 때문에 개발 기간 단축이 어렵다는 것을 알고 있다.

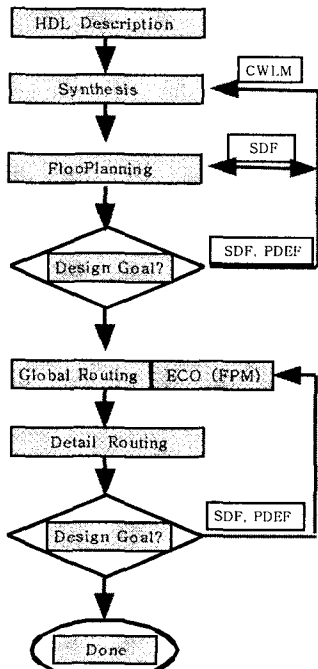
1-2. Link to Layout

<그림 2>에서는 RTL code를 Vendor wire load model을 이용하여 Synthesis 후에 FloorPlan를 하여 SDF, Cap file를 이용하여 design compiler상에서 CWLM를 생성한다. 그리고 RTL code를

recompile 한 netlist 를 생성 후에 STA 를 CWLM



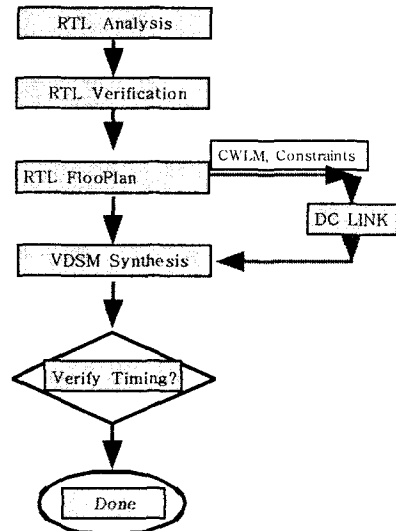
<그림-1>



<그림-2>

를 이용하여 1차 검증한다. 이렇게 검증된 netlist 를 다시 2차 FP 을 실시 후 다시 SDF back

annotation 을 실시하여 new CWLM 을 이용하여 STA 를 수행한다. 그리고 global routing 및 detail routing 을 실시하여 post simulation 을 실시한다 이때 SDF, Cap file 을 이용하여 back annotation 을 실시하여 timing met 가 안될 시에는 detail routing 된 cell 에서 SDF, Cap, PDEF (Physical Data Exchange Format)를 이용하여 LTL(Link to Layout) 을 실시한다 이때에는 cell 들의 position 정보를 가지는 PDEF(Physical Data Exchange Format)를 이용하여 정확한 ECO place 를 하게 된다. Physical synthesis 및 layout 업체에서 말하는 top level 에서 flatten 된 상태로 timing met 를 하는 concept 과 같은 의미를 가진다. 이 methodology 는 wire delay



<그림-3>

문제가 점점 대두되는 VDSM 공정과 SOC 구현 시 post iteration 을 획기적으로 줄일 수 있고 human error 를 최소화 한다.

1-3. Vendor 가 제공하는 RTL-GDS

<그림 3> EDA Vendor 중에 하나인 AVANTI 사의 Jupiter tool 의 Single-Pass-Design 의 flow 를 설명 하였다. 이 chart 의 핵심은 Layout 환경에서 Physical constraints 를 이용하여 code level 에서 layout 까지를 iteration 없이 구현 한다는 것이다 간단히 flow 를 소개 하자면 RTL code 를 input 으로 하고 chip 의 boundary condition 을 입력으로 하여 RTL analysis 및 verification 을 통하여 RTL level 에서 gate level 을 estimation 하여 FP 을 실시 하여 만들어진 CWLM 과 top level 의 timing analysis 실시 후 sub block 을 budgeting 한 constraints file 을 이용하여 DC-LINK 라는 synopsys 사의 tool 환경에서 recompile 시에 physical constraints 로 사용한다는 것이다. 요점은 physical 상에서 top down 방식으로 접근을 한다는 것이고 top level 에서의 physical constraints 를 sub block 으로 propagate 하여 budgeting 한다는데 있다

II 본론

2-1. system 환경 및 적용 과제

system은 SUN사의 Ultra 80(main memory 2 Gbyte, SWAP memory 7 Gbyte)을 사용하였고 Layout tool은 AVANTI사의 Jupiter, Milkyway, Apollo, Synthesis는 Synopsys사의 Design Compiler(V9910), Floor Plan Manager 및 Prime Time을 사용하였다. library는 삼성 반도체의 0.25 μ m 5 metal을 사용하였으며 적용 과제로는 약 4.5M gate count의 Digital TV용 chip set이다.

2-2. Jupiter tool flow 및 interface

initial constraints와 vendor의 WLM을 이용하여 quick synthesis를 진행 후 netlist를 Jupiter의 input으로 사용하고 constraints file은 initial compile 때 이용한 constraint를 사용하였다. netlist binding 후에 hierarchy FloorPlan을 실시 한 후 hard macro를 top으로 migration 후에 block diagram 별로 약 200K에서 약 700K로 sub block에 대한 soft macro를 설정하여 약 10개 이내의 soft macro 결정하였다. compile에 사용된 clock 정보와 boundary condition에 해당하는 multi cycle path, I/O delay, false path를 loading 후 top cell 및 soft macro place를 실시하였다. 각각의 soft macro에 대한 pin assign을 실시하여 block level에서의 interface되는 port에 정렬을 실시 후에 top cell들의 optimize와 routing을 실시하였다. 그리고 top에서의 timing analysis를 실시 후 DC-LINK에 필요한 sub block의 budgeting된 constraints file과 CWLM을 만들었다. 이때 만들어진 constraints file과 CWLM은 RTL code를 compile하는데 사용하였다. DC-LINK 후 만들어진 netlist는 Jupiter의 estimation된 sub block과 replace한다. 또한 이때 CWLM을 이용하여 Pre-Sim 상에서 timing met가 되면 scan insertion된 netlist를 Jupiter의 sub block과 replace한다. 그리고 다시 top level의 timing optimize를 실시 후 각각의 sub block을 flatten하여 sub block optimize를 실시한다. Sub block의 timing met가 완료되면 top level에서의 CTS(Clock Tree Synthesis)를 실시 후 global & detail routing을 실시 후 최종 sign-off를 Prime Time에서 확인 후 P & G out을 실시한다.

2-3. Jupiter flow의 문제점

RTL-GDS의 개념은 결국 두 가지의 의미로 함축된다 첫째 RTL code level에서 top-down design으로 timing met 가능 여부와, 두 번째로 Physical 정보를 이용한 one pass design이다 그러나 Jupiter flow에서의 문제점은 boundary condition에 해당하는 input constraints의 완벽한 interface가 이루어지지 않고있다 예로 false path의 propagate가 되지 않아 top down flow시 top에서의 timing analysis시 false path에 대한 sorting이 안되며 false path에 timing을 met를 위하여 많은 시간이 소요 되고, 둘째로는 Layout을 하는 Vendor의 관행이나 CMP(Chemical Mechanical Polishing)문제와

Power 문제로 layout 시에는 memory, PLL, ADC, DAC 등에 관한 hard macro는 CORE의 PAD 근처에 배치 하게 되는데 Jupiter는 memory의 top level에서 flatten 시에 soft macro 안에서 migration된 memory cell들의 port들이 soft macro 상에서 virtual port로 인식된다 따라서 DC LINK를 실시할 때 사용되는 budgeting constraints file이 design의 inout port로 존재하지 않기 때문에 DC-LINK 시에 error를 유발하는 문제를 야기하고 있다. 또한 세 번째로는 Jupiter 환경에서는 scan cell 및 chain을 insertion이 불가능하다는 것이다. 물론 DC-LINK 후에 sub block의 replace를 실시할 때 scan insertion이 될 수 있다. 그러나 이때의 문제점은 soft macro의 area와 FP의 변경이 불가피 할 수도 있게 되며 다시 한번 DC-LINK를 필요 시에는 compile을 가능한 한 피해야 한다. 왜냐 하면 compile 시에는 scan cell 및 insertion을 다시 하기 때문이다 아니면 optimize만을 실시하게 되면 scan cell들에 관한 정보는 변하지 않아 그대로 사용이 가능하다.

2-3. Physical synthesis

Physical compile은 Physical 정보를 이용하여 code level에서 compile 및 optimize를 의미 하며 physical constraints에는 wire delay, capacitance, cell의 location인 PDEF 3개로 구성된다 기존의 FP 정보를 이용하여 compile을 한다. physical compiler는 input으로 physical library와 PDEF를 이용하여 compile을 실시하고 동일 환경에서 FP을 실시하여 timing met를 확인 후 PDEF 정보와 physical constraints를 이용하여 optimize를 실행한다. 물론 PDEF 정보를 가지려면 code level에서의 initial compile된 netlist는 필요하다 결국 CWLM, FP 정보 Physical constraint를 이용하는 면에서는 LTL flow에서 SDF, PDEF를 사용하여 FPM를 이용하는 것과 큰 차이가 없음을 알 수 있다. 단 동일한 tool 환경에서 compile 및 timing met가 가능하며 FP이 가능하다는 것이 큰 장점 이랄 수 있다.

2-4. LTL flow 적용

general한 constraint file을 input으로 code를 compile한다 이때는 물론 DFT를 위한 test ready compile을 수행한다 1차로 만들어진 netlist를 AVANTI사의 PlanetPL 상에서 FP을 실시 하여 region 및 grouping을 실시하였다 그리고 SDF, capacitance를 추출하여 design compile 상에서 CWLM을 만든 후 RTL code를 recompile 후 compile시 사용한 CWLM을 이용하여 STA를 실시하였다. 그리고 이 netlist를 다시 back annotation하여 2차 FP를 실시하였다. 이때는 STA 결과에 따라 FP을 변경 하였다. 2차로 만들어진 SDF 및 CAP file을 이용하여 2차 CWLM을 이용하여 RTL code를 recompile을 하고 cell depth가 긴 3개의 block에 대해서는 code수정을 하여 수정된 code는 re-compile을 하여 2차 CWLM을 이용하여 STA를 실시 하였다. 그리고 다시 netlist를 이

용하여 FP을 한 후 global routing을 실시 하고 SDF, Cap, PDEF 정보를 back annotation하여 FPM 상에서 re-optimize를 실시한 후 top level에서의 new PDEF 및 new netlist를 ECO place를 실시 하였다. 그러나 문제점은 gate count가 4M 이상이다 보니 top level에서 optimize나 compile시에 out of memory가 발생하여 re-optimize와 compile을 top에서 할 수 없다는 것이다. 물론 과거에 1.3M급의 Format convert chip-set에서는 전혀 문제가 되지 않았지만 4M 이상급의 SOC chip에서는 top level에서의 handling이 불가능 하였다. 따라서 top level에서 characterize를 실시 후 re-optimize를 진행하여 new cluster를 update하여 새로운 top level의 PDEF를 만들었으며 ECO를 진행 시 사용하였으며 physical constraints file은 Prime time 상에서 budgeting하여 sub top level에서 사용 하였다.

III. 결과

RTL code를 initial compile 후에 FP하여 만들어진 CWLM을 사용하여 recompile 하면 약 20% ~ 30%의 area가 증가됨을 알 수 있었다. 이 netlist를 이용하여 Jupiter에서 top을 timing analysis 시 약 24hour이 소요 되었고 이때의 문제점은 Jupiter 환경 상에서 STA를 실시 중에 false path가 tool 상에서 interface되지 않아 오랜 수행 시간을 가져왔고 또한 false path에 대한 timing slack 분석에 많은 시간이 소요 되었다. boundary condition file들이 Jupiter 환경으로 완벽하게 호환 되지 않았다. 결국 sub block에 해당하는 soft macro에 top의 constraints를 sub block에 budgeting 후 만들어진 budgeting script에 false path 및 I/O delay가 정확하게 budgeting 되지 않아서 DC-LINK가 완벽하게 지원 되지 않았다. 또한 hard macro를 core top으로 migration 시에 memory가 top으로 올라 오면서 soft macro에 virtual port들이 생겨서 DC-LINK 시에 error를 유발 시켰다. 이 문제는 hard macro cell들을 top level에서 따로 module화하면 virtual port 문제는 해결이 가능 할 것이다. LTL을 적용 시에 3회의 FP을 실시 하였고 이때의 insertion cell들은 1.5M glue logic cell 기준으로 약 300K gate area 정도의 증가를 가져 왔고 이때 생성된 PDEF를 가지고 FP을 실시 할 때 Layout 상에서 region과 group area 안으로 들어가는 new cell과 resizing cell로 인하여 FP의 변경이 불가피 하였다. 또한 많은 sub block에서 slack이 발생하여 해당 block을 characterize 하는데 많은 시간이 소요 되었다. 최초의 FP 후 만들어진 CWLM을 이용한 timing violation을 확인 시 약 3.5ns의 set up violation이 발생 하였고 FP 후 SDF를 이용하여 re-optimize 실시 후 back annotation하여 STA를 확인 시 개선된 slack은 1.5ns 이하였다. block 간 interface net들은 excessive load 형태의 cell delay가 커지는 문제를 가지고 있었다. 또한 WLM을 사용 할 때 와 SDF를 사용하여 발생하는 long path의 net의 차이는

여전히 존재 하였다. debugging의 형태는 top에서 optimize 및 PDEF를 이용하여 re-optimize를 가지고 실시하였고 CWLM에 대한 over constraints를 주어서 해결 하였다.

IV. 결론

Layout 이후 발생하는 timing violation을 fix 하는데 사용한 CWLM, SDF, PDEF는 VDSM 공정을 사용한 SOC 개발을 위하여 반드시 사용 되어야 할 parameter임을 확인했다. Physical 정보를 이용한 SOC 개발은 이미 0.35 um 공정이 채택된 1.3M 이상의 SOC chip 개발에서 입증 이 되었고 wire delay의 비중이 높아지는 0.35um 이하 공정에서 혹은 design size가 수백만 gate로 되는 과제에서는 필수적으로 사용 되어야 할 methodology임을 알 수 있었다. 하지만 이 방법은 Post-Sim 이후의 timing debugging 방법이고 RTL code 상에서 synthesis는 반드시 Physical 정보를 사용해야 한다는 것이다. VDSM 공정을 이용한 SOC 개발을 위해 Single Pass Design, Physical Synthesis, top level optimize 등의 이름으로 개발되고 있는 tool들은 아직까지는 CWLM 정도의 지극히 제한적인 parameter만 이용되고 있으며 SOC chip의 RTL-GDS 구현을 위해서는 RTL code level에서 physical 정보를 top down 방식으로 완벽하게 이용 가능하며 tool 간의 iteration 없이 동일 환경 안에서 구현되어야 할 것이며 앞으로도 많은 시간이 필요함을 알 수 있었다.

- [1] Pran Kurup, Taher Abbasi "Logic Synthesis using synopsys", KLUWER ACADEMIC PUBLISHRES 1995
- [2] Floorplan Manager success in deep sub-micron Design., SYNOPSYS
- [3] "RTL Analysis and VDSM Synthesis" guide book., AVANTI, 1999 Spring Technology Tour
- [4] From RTI to GDS II Technology and case Study., CADENCE
- [5] "SoC Design Challenges and Physical Synthesis", SYNOPSYS, Inc, The Millennium Korea Spring Tech Seminar