CSEE 6861 CAD of Digital Systems
Handout: Lecture #12
4/14/16

Prof. Steven M. Nowick
nowick@cs.columbia.edu

Department of Computer Science (and Elect. Eng.)
Columbia University
New York, NY, USA

Overview of Architectural-Level Synthesis
(= High-Level Synthesis)

Key Synthesis/Optimization Steps: at 3 Levels

1. Architectural Synthesis (also, "High-Level Synthesis" [HLS])

Starting point: behavioral system specification

Steps: ***scheduling, resource allocation (sharing) and binding***

Outcome: register-transfer level (RTL)] optimized design
for block-level datapath + FSM controller specification

2. Logic Synthesis

Steps:

sequential synthesis: *FSM optimization*

combinational synthesis: *(i) 2-level logic minimization, (ii) multi-level logic optimization*

technology mapping: *optimal mapping of gates to VLSI "library" cells*

Outcome: mapped gate-level circuit

3. Physical Design

Steps: ***circuit partitioning, chip floorplanning, place-and-route ("P&R")***
... *+ late timing correction/optimizations, etc.*

Outcome: complete chip layout ➔ ready for fabrication

#3

Architectural Synthesis

High-Level Specification: Differential Equation Solver *(diff-eq)* Custom Unit

```
diffeq {
        read ( x, y, u, dx, a );
        repeat  {
                xl  =   x + dx;
                ul  =   u - ( 3 * x * u * dx ) - ( 3 * y * dx );
                yl  =   y + u * dx;
                c   =   xl <  a;
                x = xl ; u = ul ; y = yl;
                }
        until ( c );
        write ( y );
}
```

p. 19, DM book

Figures courtesy of: G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill (1994)

#4

Architectural Synthesis

Target Micro-Architecture: Register-Transfer Level

Version #1

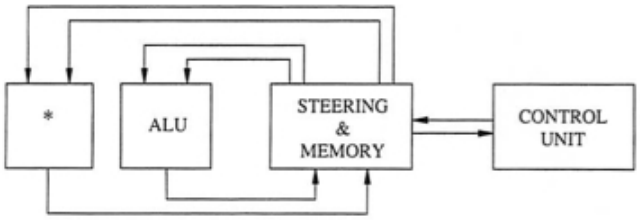


FIGURE 1.12
Example of structural view at the architectural level.

p. 19, DM book

#5

Architectural Synthesis

Target Micro-Architecture: Register-Transfer Level

Version #2

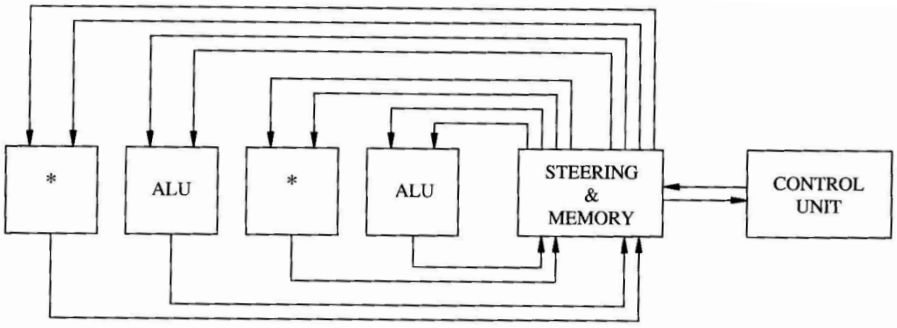


FIGURE 1.14
Alternative example of structural view at the architectural level.

p. 22, DM book

#6

Architectural Synthesis

More Detailed Target Micro-Architecture: RTL Level

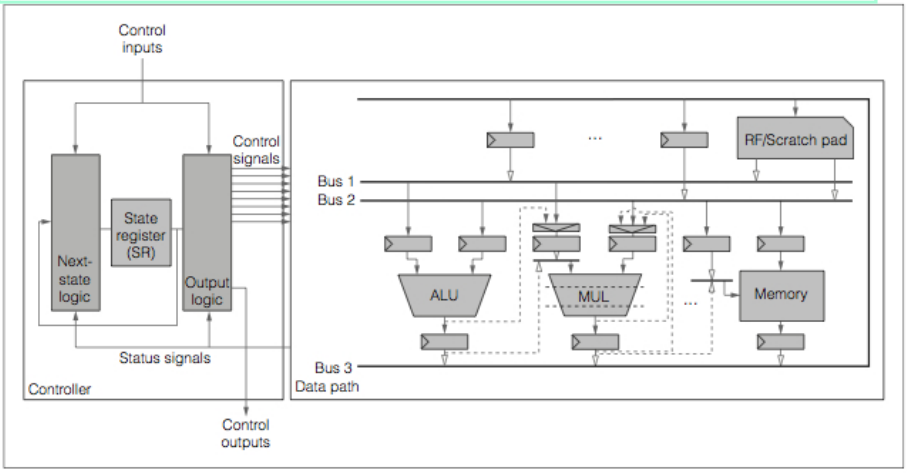


Figure 2. Typical architecture.

Courtesy of: P. Coussy, D.D. Gajski, M. Meredith and A. Takach,
"An Introduction to High-Level Synthesis", IEEE Design & Test of Computers (July/Aug. 2009)

#7

Architectural Synthesis: Deriving a CDFG Spec

High-Level Specification: Differential Equation Solver *(diff-eq)* Custom Unit

```
diffeq {
        read ( x, y, u, dx, a );
        repeat  {
                xl  =   x + dx;
                ul  =   u - ( 3 * x * u * dx ) - ( 3 * y * dx );
                yl  =   y + u * dx;
                c   =   xl <  a;
                x = xl ; u = ul ; y = yl;
                }
        until ( c );
        write ( y );
}
```

initial pseudo-code

$$xl = x + dx;$$
$$ul = u - (3 * x * u * dx) - (3 * y * dx);$$
$$yl = y + u * dx;$$
$$c = xl < a;$$

extract pseudo-code of inner loop body

p. 121, DM book

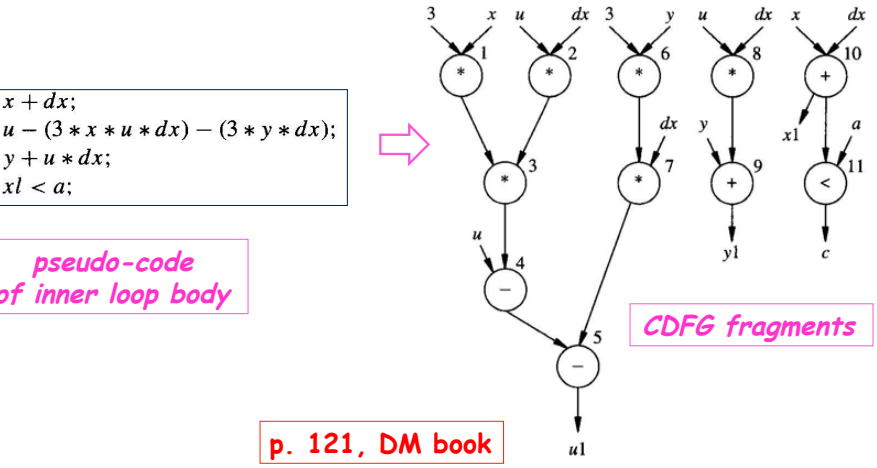
#8

Architectural Synthesis: Deriving a CDFG Spec

High-Level Specification: Differential Equation Solver *(diff-eq)* Custom Unit

```
xl = x + dx;
ul = u − (3 ∗ x ∗ u ∗ dx) − (3 ∗ y ∗ dx);
yl = y + u ∗ dx;
 c = xl < a;
```

pseudo-code of inner loop body



CDFG fragments

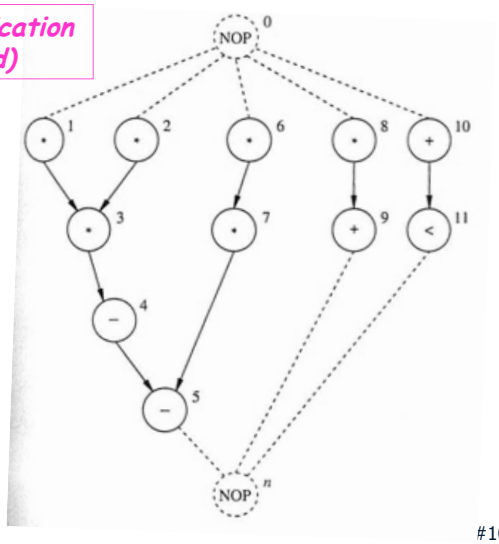
p. 121, DM book

#9

Architectural Synthesis: Scheduling

Unscheduled Control-Dataflow Graph (CDFG): *diff-eq*

Final unscheduled CDFG specification (constant operands omitted)

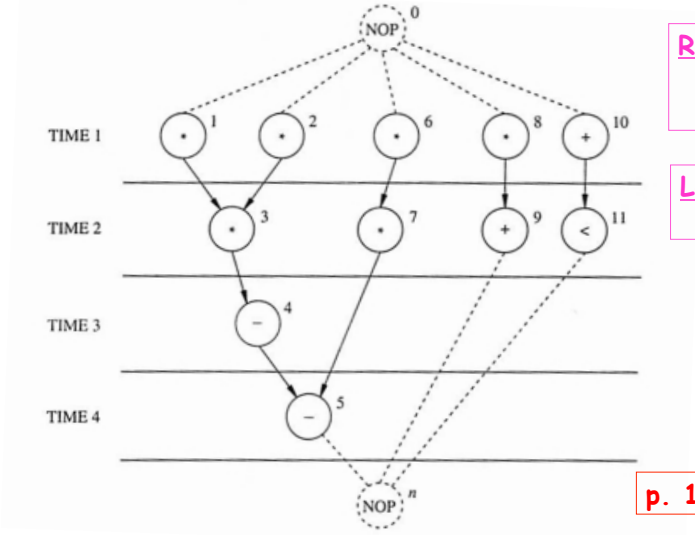


p. 147, DM book

#10

Architectural Synthesis: Scheduling

Scheduled CDFG: minimum-latency



Resources:
4 MULT units
2 ALU's

Latency:
4 cycles

p. 148, DM book

#11

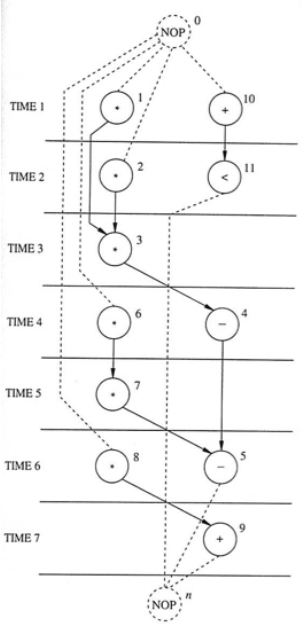
Architectural Synthesis: Scheduling

Scheduled CDFG: minimum-area
= resource-constrained (RC)

Resources:
1 MULT unit
1 ALU

Latency:
7 cycles

p. 149, DM book

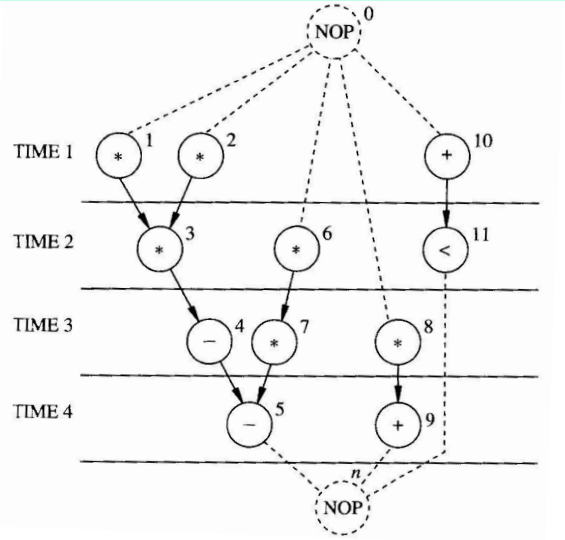


#12

Architectural Synthesis: Scheduling

Scheduled CDFG: 2-tiered cost function

min-area *(secondary cost)*, under min-latency constraints *(primary cost)*



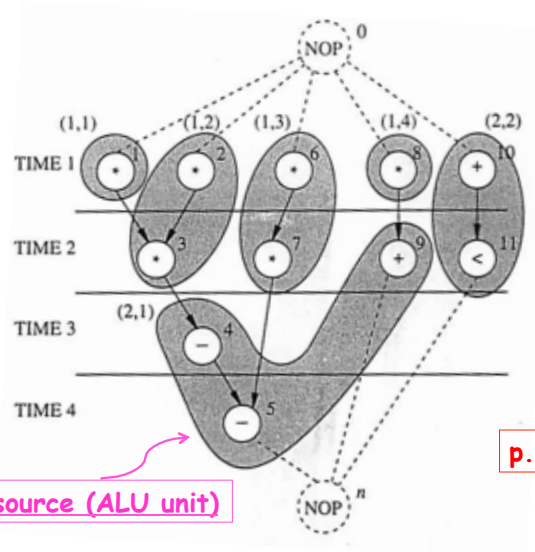
Resources:
2 MULT units
2 ALU's

Latency:
4 cycles

p. 201, DM book

#13

Architectural Synthesis: Resource Binding



1 shared resource (ALU unit)

p. 152, DM book

#14