CSEE 6861 CAD of Digital Systems
Handout: Lecture #6
2/25/16

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

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

Multi-Level Optimization: Overview of Basic Transforms

(taken from Giovanni De Micheli, "Synthesis and Optimization of Digital Circuits" (McGraw Hill), ch. 8.2)

Basic Representation #1: Logic Network

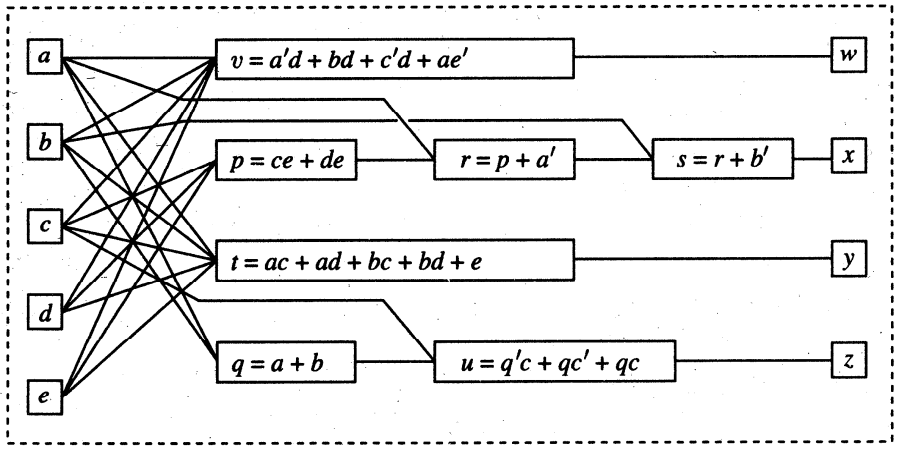


Fig. 8.3(a)

#3

Basic Representation #2: Logic Network Graph

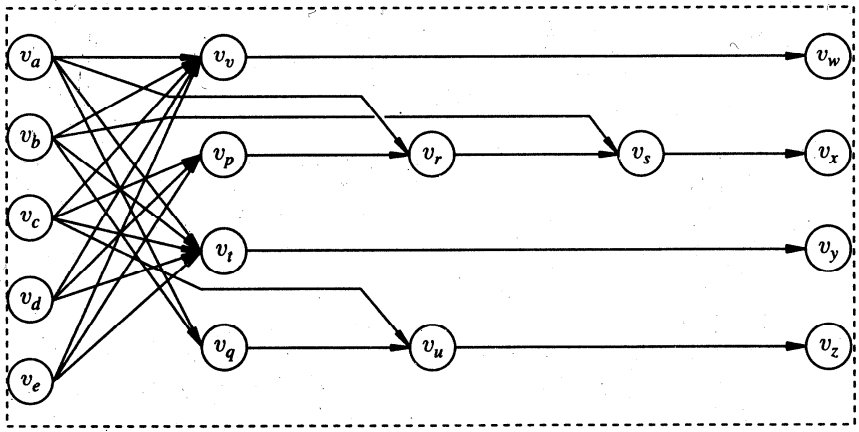


Fig. 8.3(b)

#4

Comparison: Logic Network vs. Logic Network Graph

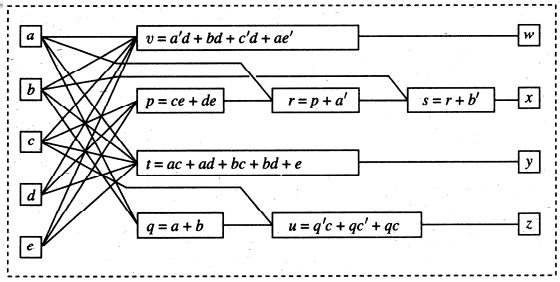


Fig. 8.3(a)

Logic network

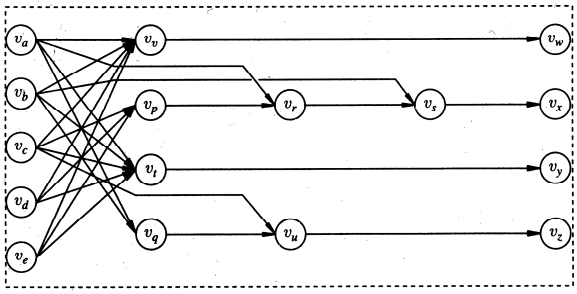


Fig. 8.3(b)

Logic network graph

Multi-Level Optimization Example: Initial Design

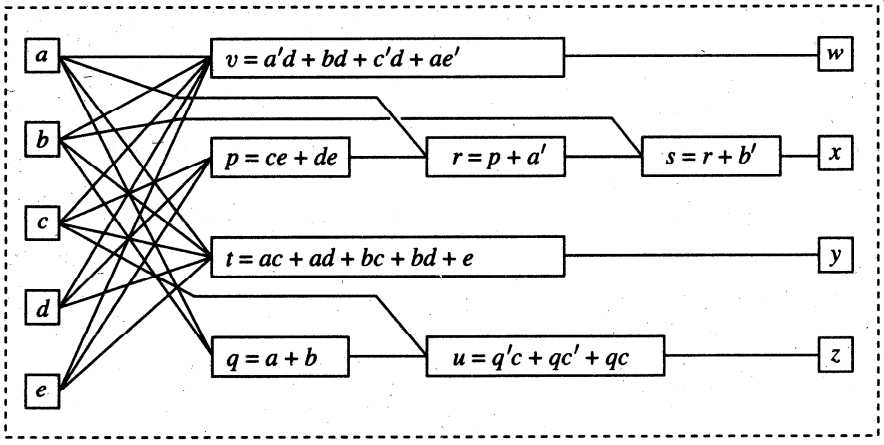


Fig. 8.3(a)

Transform #1: ELIMINATE (collapse)

Eliminate a node: "collapse" network structure

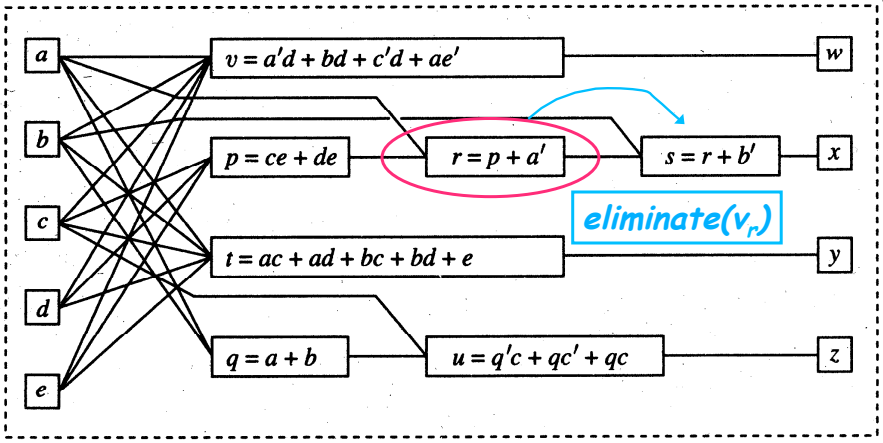


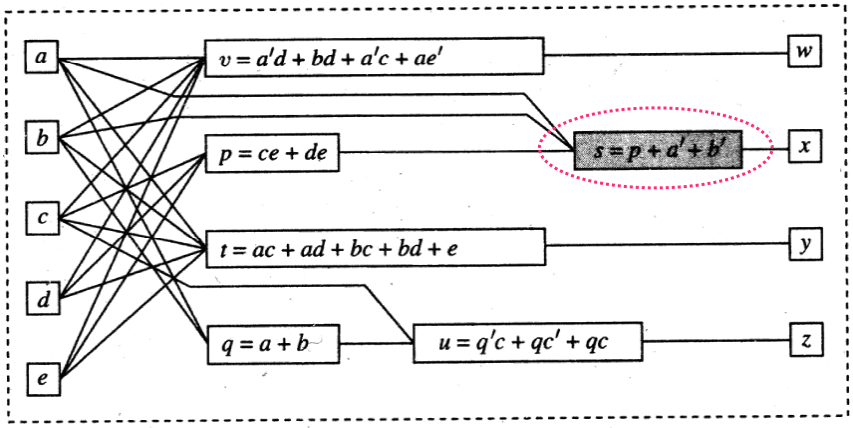
Fig. 8.3(a)

before

#7

Transform #1: ELIMINATE (collapse)

Eliminate a node: "collapse" network structure



after

#8

Transform #2: DECOMPOSE

Break 1 larger node into several smaller nodes

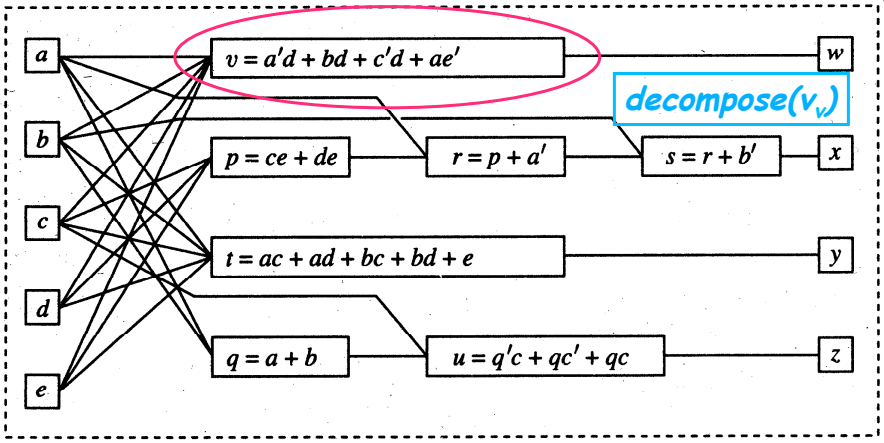


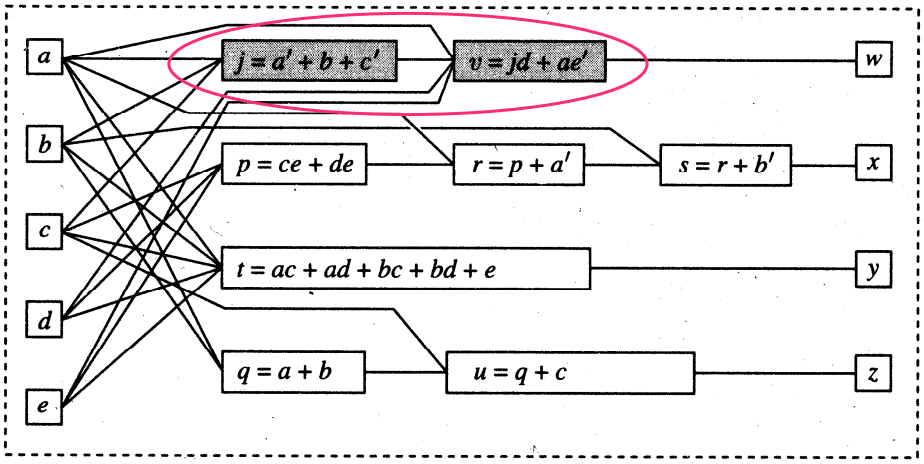
Fig. 8.3(a)

before

#9

Transform #2: DECOMPOSE

Break 1 larger node into several smaller nodes



after

#10

Transform #3: EXTRACTION

Create/extract "common subexpression" for 2 or more nodes

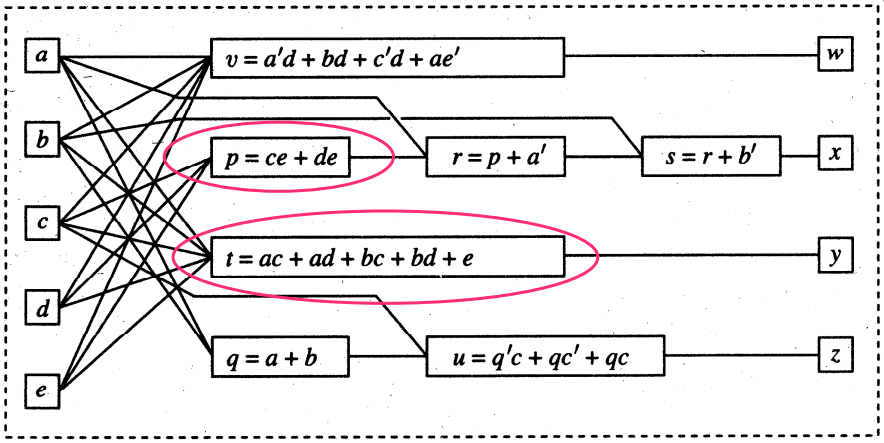
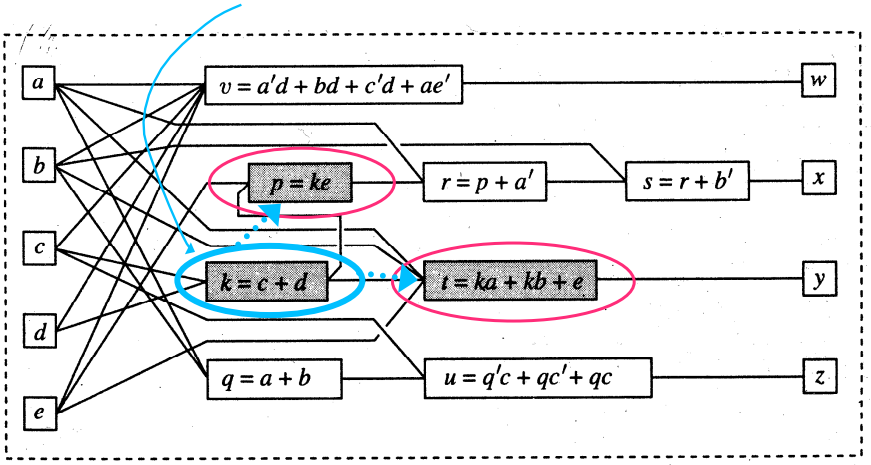


Fig. 8.3(a)

before

Transform #3: EXTRACTION

Create/extract "common subexpression" for 2 or more nodes



after

Transform #4: SIMPLIFICATION

Perform optimization (usually Boolean) within a single node

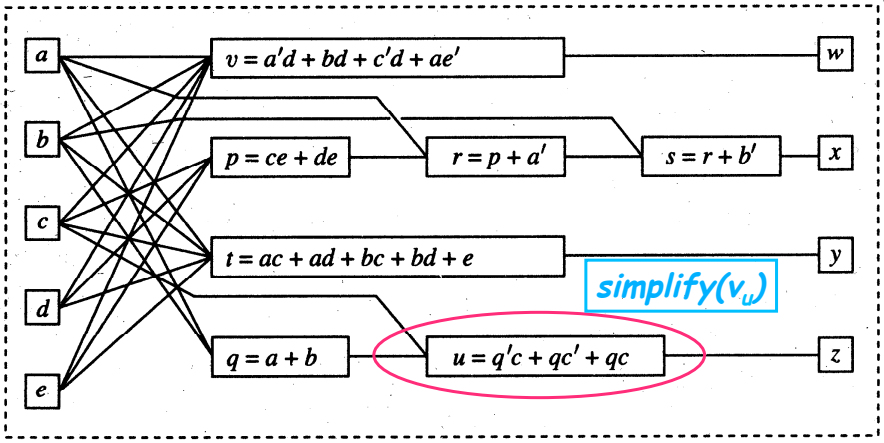
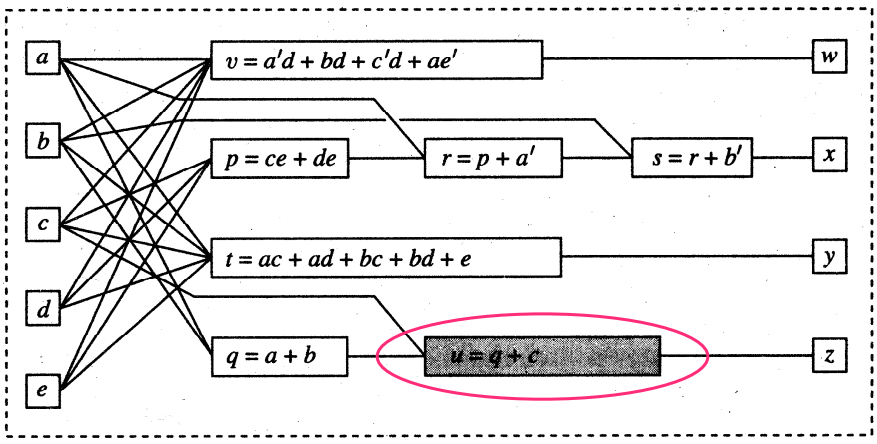


Fig. 8.3(a)

before

Transform #4: SIMPLIFICATION

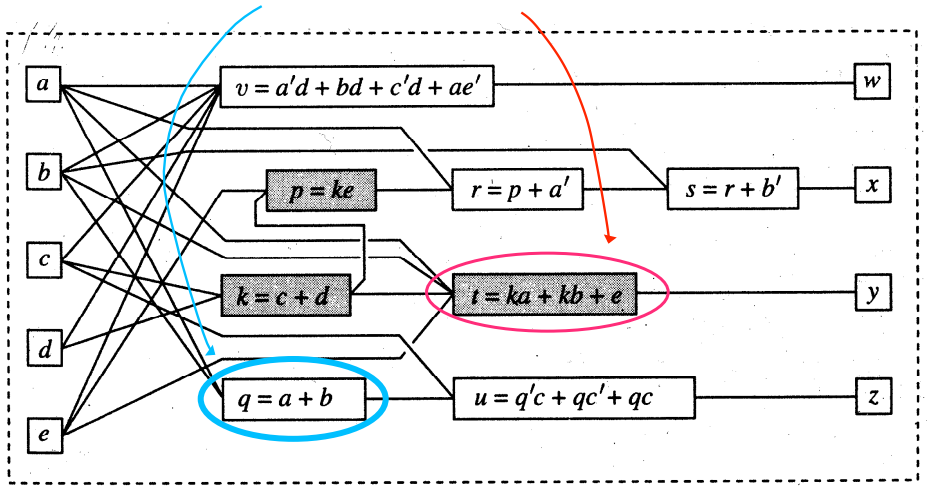
Perform optimization (usually Boolean) within a single node



after

Transform #5: SUBSTITUTION

Find an existing "common subexpression" for 1 or more nodes

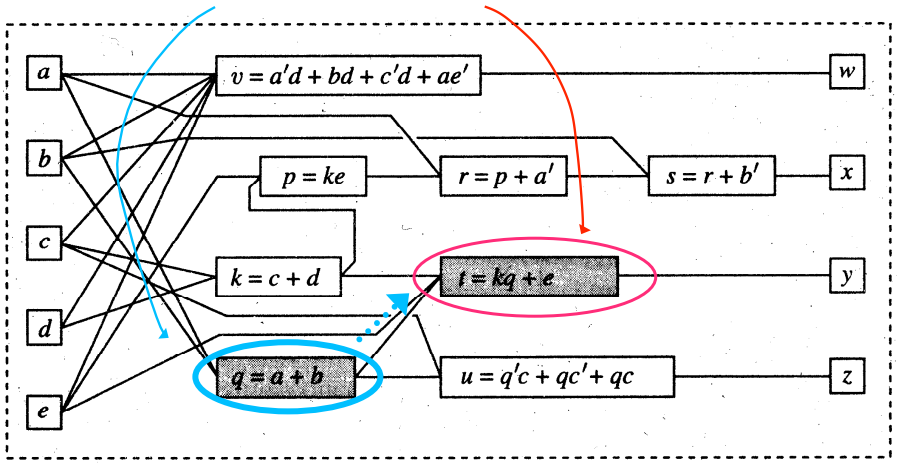


before

#15

Transform #5: SUBSTITUTION

Find an existing "common subexpression" for 1 or more nodes



after

#16