Common Sub-expression Elimination

Payman Samadi

Supervisor: Dr. M. Ahmadi

Outline

- Introduction
- Horizontal Common Sub-expression Elimination
- Vertical Common Sub-expression Elimination
- Design Procedure
- Design with Genetic Algorithm
- Example
- Results
- Future Works

Common Sub-expression Elimination

- When a portion of an expression (sub-expression) occurs more than once, it can be calculated once and the result can be used further.

- Sub-expression can be any bit patterns within the CSD coefficients. $(101, 10\bar{1}, 1001)$
  - Horizontal Sub-expression
  - Vertical Sub-expression

Horizontal Sub-expression Elimination

- Sub-expressions can be found and eliminated horizontally:

$$y = (1010101)_x$$

$$y = x + x << 2 + x << 4 + x << 6$$

$$y = x + x << 2 + (x + x << 2) << 4$$

$$s = x + x << 2 \Rightarrow (101)$$

$$y = s + s << 4$$

Vertical Sub-expression Elimination

- When coefficients are staked, sub-expressions can be found vertically:

$$y = (1\,0\,0\,\bar{1}\,0\,1)x[0] + (1\,0\,1\,0\,0\,1)x[-1]$$

$$y = x[0] - x[0] << 2 + x[0] << 5 + x[-1] + x[-1] << 3 + x[-1] << 5$$

$$s = x[0] + x[-1]$$

$$y = s + s << 5 - x[0] << 2 + x[-1] << 3$$

Design Procedure

- Stack Coefficients Vertically
- Creating Graph of Vertices: V_{id}
- Creating Partial Identification Graph: G'_{id}
  - $G'_{id} = (V_{id}, E'_{id})$ $E'_{id} = E_h + E_v$
- Creating the Final Edge: $E_{id} = E'_{id} - E_{unique}$
- Creating Identification Graph: $G_{id} = (V_{id}, E_{id})$
- Creating the Search graph: S

Design Procedure

- Suppose a filter with following coefficients:

$$c_0 = 1010\bar{1}001 \qquad c_1 = 10000101$$

- Creating the vertical stack:

c_0	$1\,0\,1\,0\,\bar{1}\,0\,0\,1$
c_1	$1\,0\,0\,0\,0\,1\,0\,1$

Design Procedure

- Graph of Vertices: $V_{id} = \{V_1 \ldots V_7\}$

c_0	$10101\bar{1}001$
c_1	10000101

Vertex (Non-zero digit)	Digit Polarity	Coefficient	digit Position
V_1	+1	0	8
V_2	+1	0	6
V_3	-1	0	4
V_4	+1	0	1
V_5	+1	1	8
V_6	+1	1	3
V_7	+1	1	1

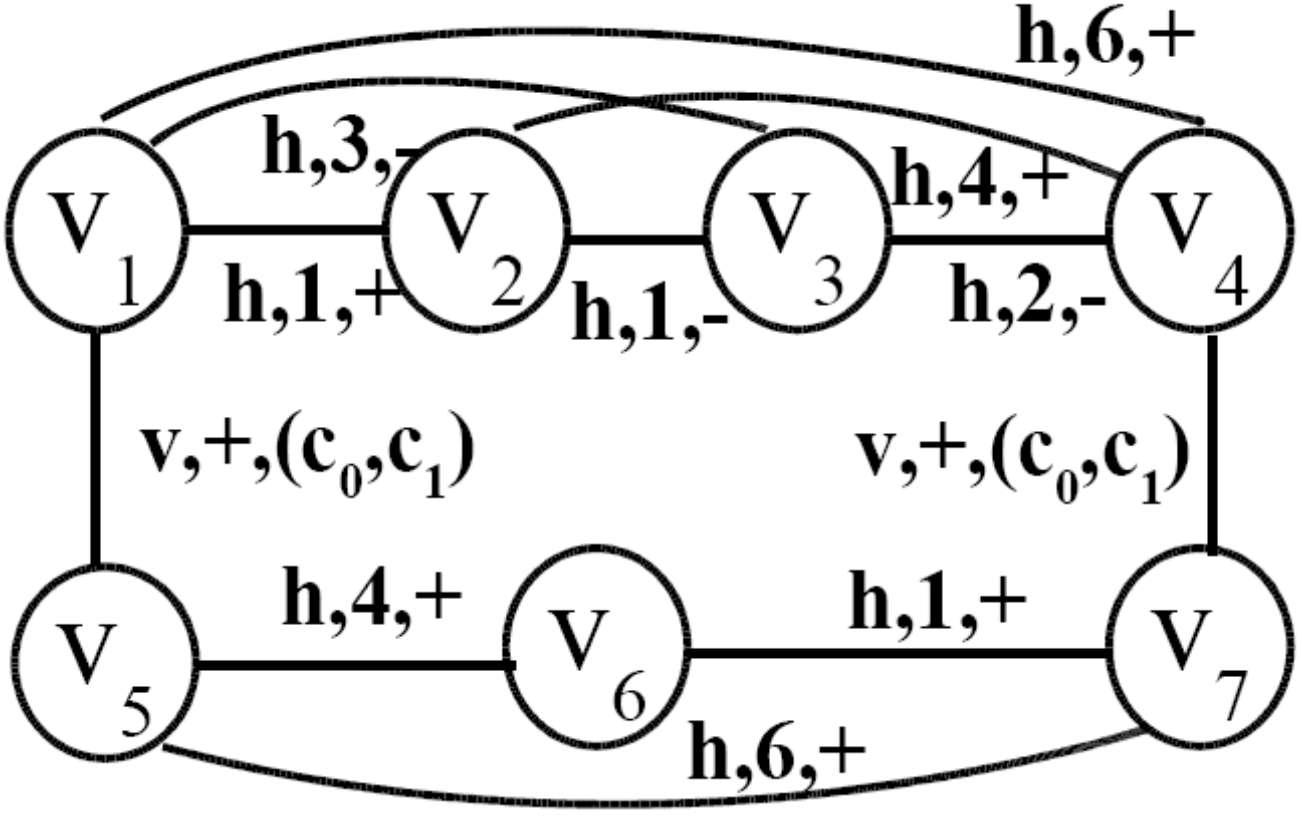
Design Procedure

- Partial ID Graph: G'_{id}

c_0	$1\,0\,1\,0\,\bar{1}\,0\,0\,1$
c_1	$1\,0\,0\,0\,0\,1\,0\,1$

$$G'_{id} = (V_{id}, E'_{id})$$

$$E'_{id} = E_h + E_v$$



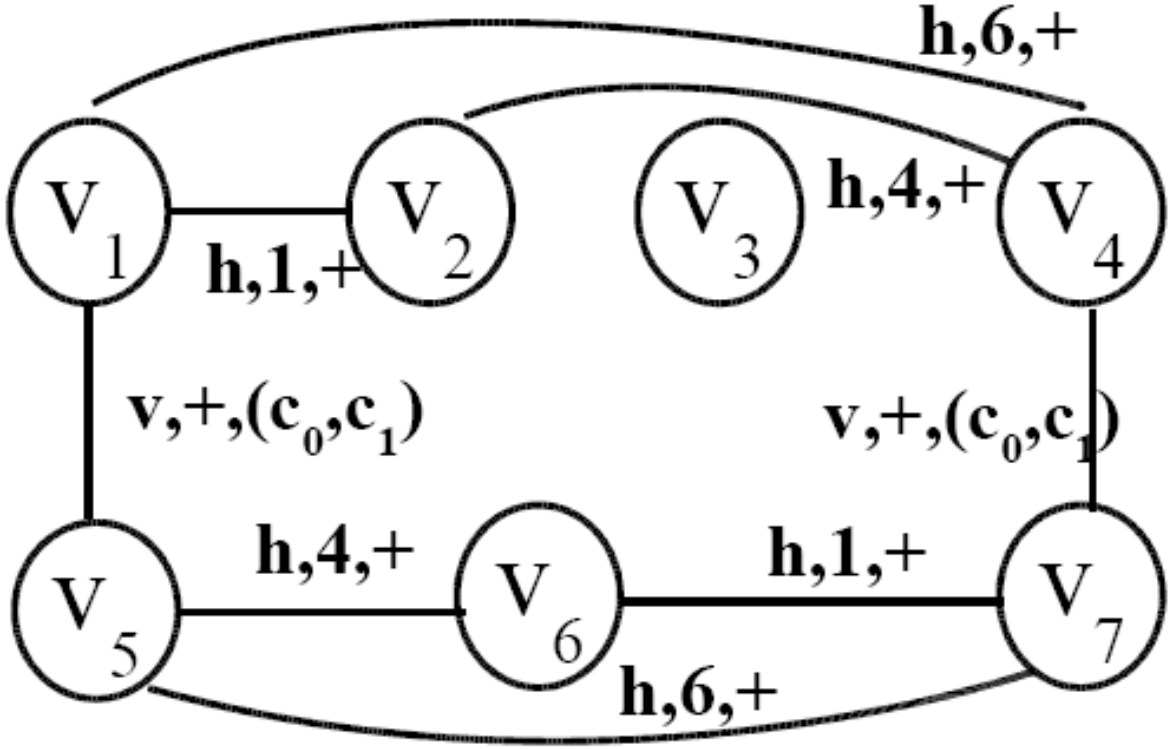
Edge list E'_{id} with Edge Properties:

Edge	(V_a,V_b)	Type	Polarity	Length	(C_1,C_2)
1	(1,2)	h	+	1	----
2	(1,3)	h	-	3	----
3	(1,4)	h	+	6	----
4	(1,5)	v	+	----	(c_0,c_1)
5	(2,3)	h	-	1	----
6	(2,4)	h	+	4	-----
7	(3,4)	h	-	2	-----
8	(4,7)	v	+	----	(c_0,c_1)
9	(5,6)	h	+	4	----
10	(5,7)	h	+	6	----
11	(6,7)	h	+	1	----

Design Procedure

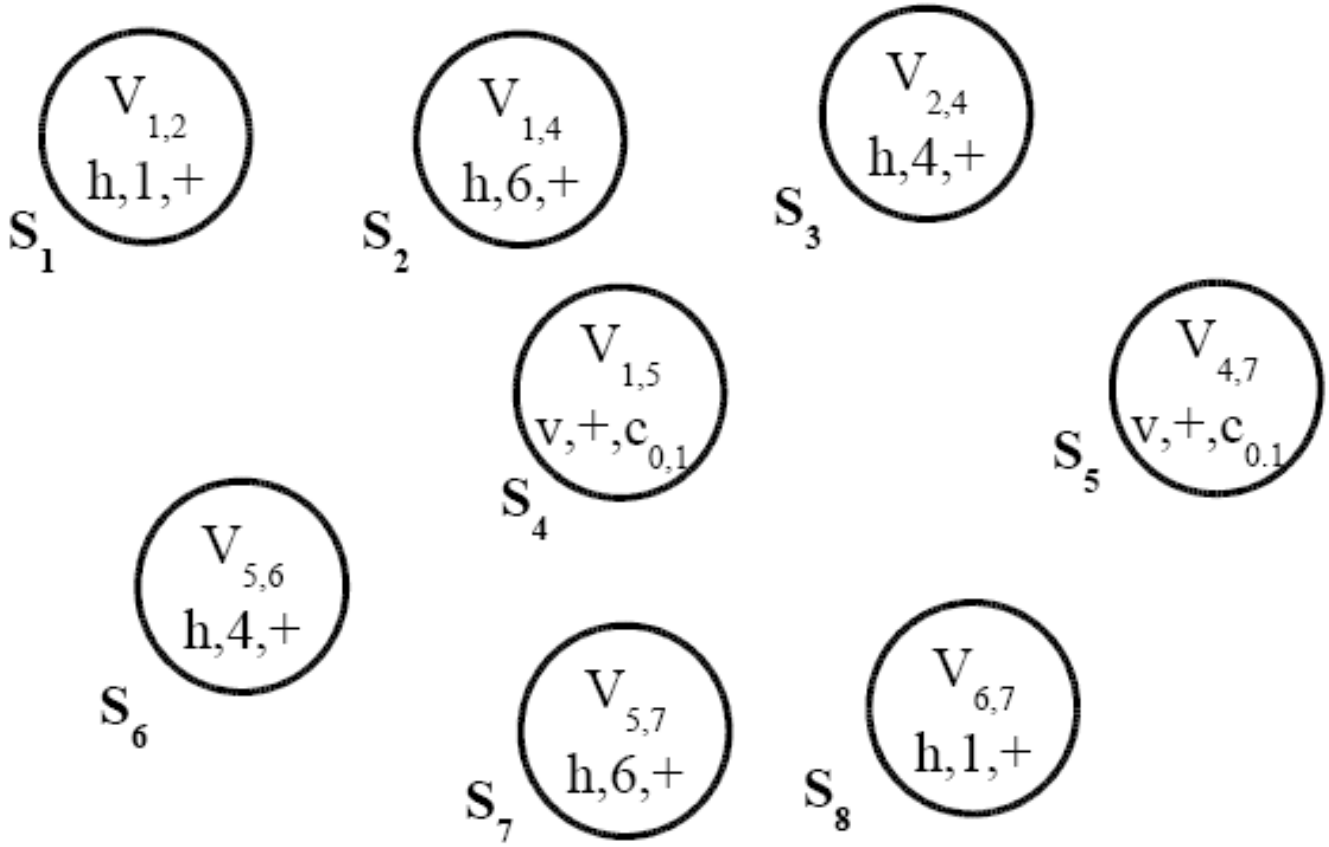
- Completed ID Graph: G_{id}

$$E_{id} = E'_{id} - E_{unique}$$



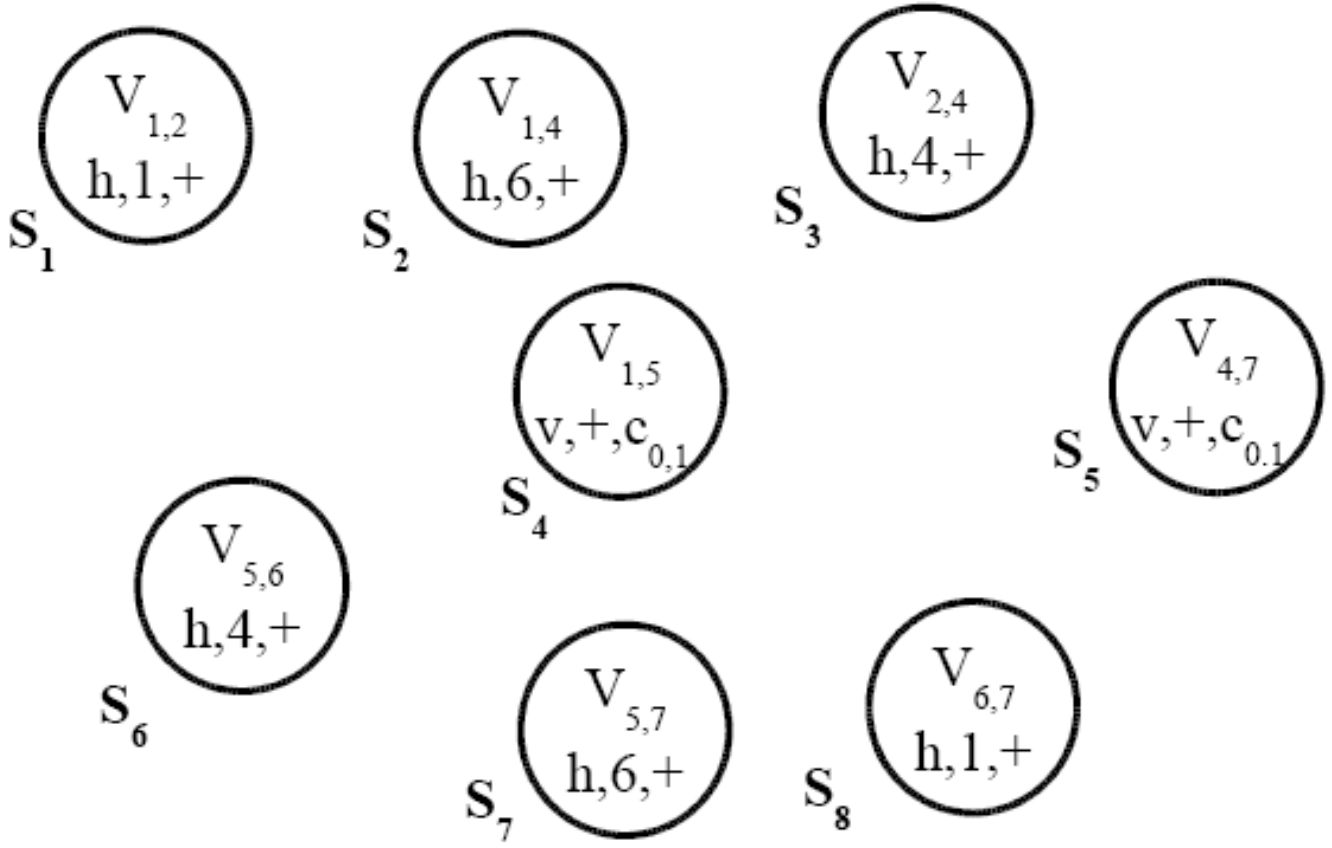
Design Procedure

- Search Graph:



Design Procedure

- Hamiltonian Walk: S_7, S_6, S_3, S_5, S_8, S_4, S_1, S_2.



Design Procedure

- ID Graph Vertex Availability Table after (S6, S3) Elimination:

V_{id} 1	Available
V_{id} 2	Not Available
V_{id} 3	Available
V_{id} 4	Not Available
V_{id} 5	Not Available
V_{id} 6	Not Available
V_{id} 7	Available

Design with Genetic Algorithm

- Fitness Function:

$$fitness : \sum_{i=1}^{n} (OC_i - 1) \qquad (If\ OC_i > 1\ else\ 0)$$

  - OC: Occurrence Count: Number of Elimination

Hamiltonian Walk (Chromosomes):

S7, S6, S3, S5, S8, S4, S1, S2

Example

- 10^{th} order FIR filter with 16bit CSD digits and maximum 3 non-zero digits:

a0	0	-3H	0	0	0	-3H	0	0	0	0	0	0	0	-1	0	0
a1	0	0	-5H	0	1	0	0	0	-5H	0	0	0	0	0	0	0
a2	0	0	0	0	0	+6V	0	0	+9V	0	+11V	0	0	0	0	0
a3	0	0	0	0	+1H	0	+1H	0	0	-1	0	0	0	0	0	0
a4	0	0	0	0	0	0	+1H	0	+1H	0	-1	0	0	0	0	0
a5	0	0	0	0	0	+6V	0	`0	+9V	0	+11V	0	0	0	0	0
a6	0	0	0	0	0	0	-5H	0	-1	0	0	0	-5H	0	0	0
a7	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
a8	0	0	0	0	0	0	0	0	-5H	0	0	-1	0	0	-5H	0
a9	0	0	0	0	0	0	0	0	0	0	-3H	0	0	0	-3H	0
a10	0	0	0	0	0	0	0	0	0	0	-3H	0	-1	0	-3H	0

Improvement Table

- 10th order FIR filter with 16bit CSD digits:

Filter (by non-zero digit count)	Original Addition Count	Reduction (additions)	Additions Required After Elimination	Reduction (%)
6	45	14	31	31.1%
5	36	10	26	27.7%
4	40	13	27	32.5%
3	29	7	22	24.1%
2	20	6	14	30.0%

Future Works

- Using Immune Programming as an optimization function
- Trying Different Algorithms for Elimination