OPTIMIZING COARSE-GRAINED UNITS IN FLOATING POINT HYBRID FPGA

Chi Wai Yu[1], Alastair M. Smith[2], Wayne Luk[1], Philip Leong[3], Steven J.E. Wilton[2]

[1]Dept of Computing
Imperial College London,
London
{cyu,wl}@doc.ic.ac.uk

[2]Dept of Electrical
and Computer Engineering,
University of British Columbia,
Vancouver, B.C., Canada
{alastairs, stevew}@ece.ubc.ca

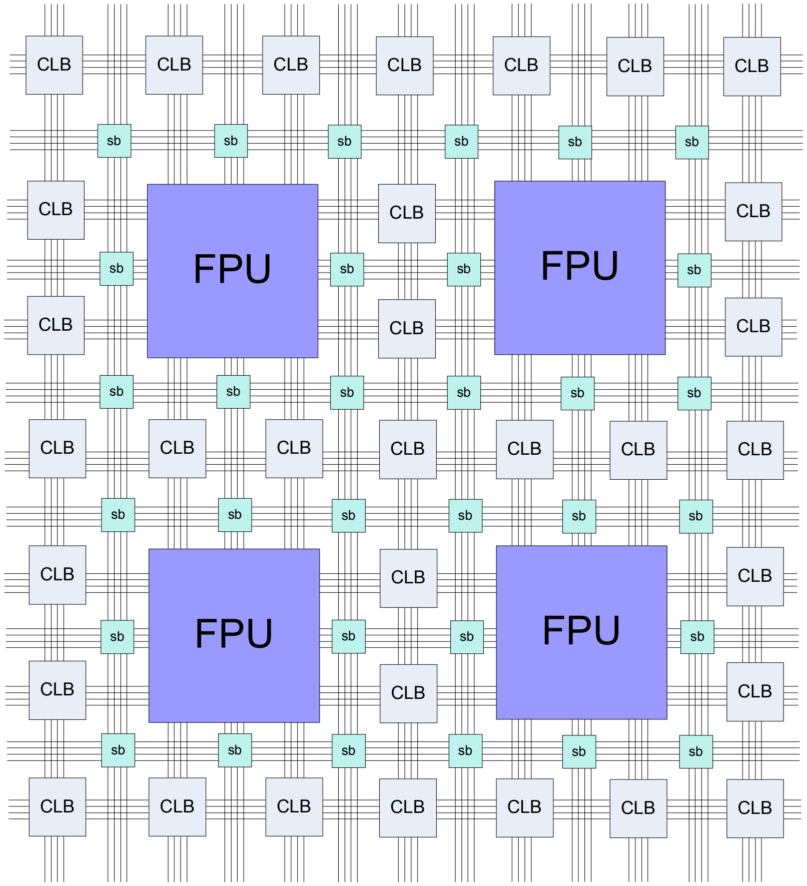
[3]Dept of Computer
Science and Engineering,
Chinese University of
Hong Kong
phwl@cse.cuhk.edu.hk

Outline

1. Motivation and background
2. Contributions
3. Methodology
4. Study of FPUs architecture
5. Quantitative system level analysis
6. Conclusion

1. Motivation

- Aim: to optimize coarse-grained Floating Point Units (FPUs)
- Internal optimization: area, speed and utilization of individual FPU
- System level optimization:
  - impact of density and flexibility of FPUs on the system
  - area, speed and routing resources



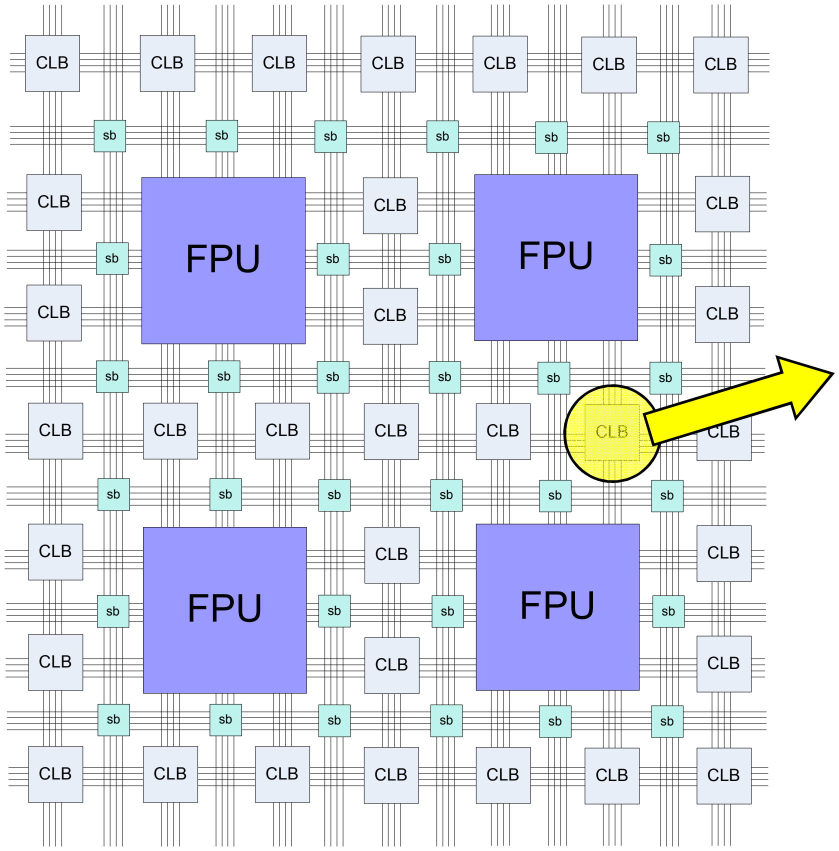
Background

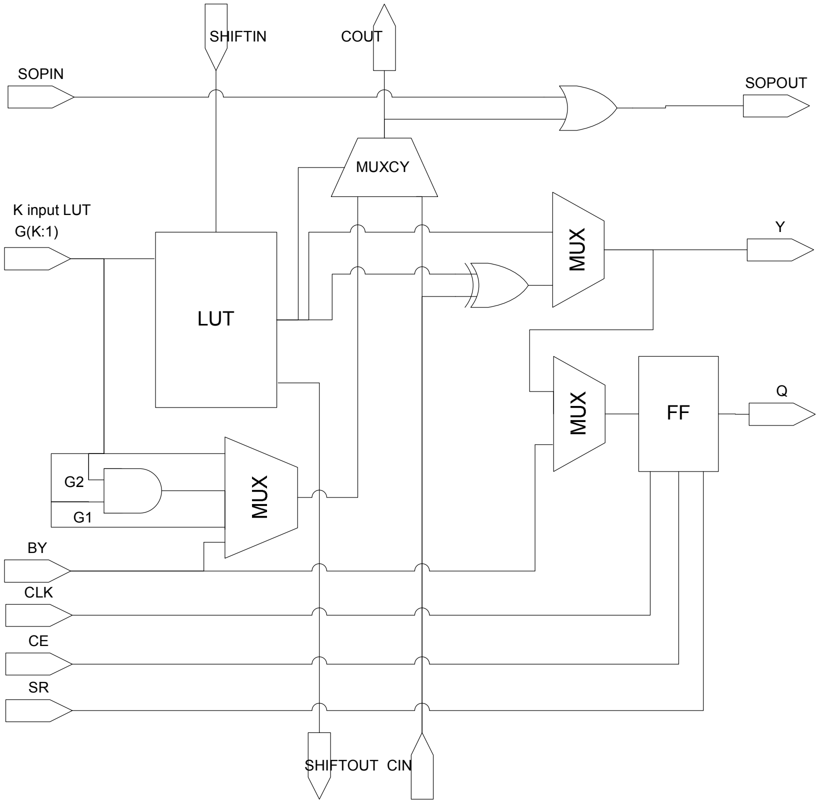
- Coarse-grained blocks, e.g. DSP48 in XC4V devices
  - improve area and delay of fine-grained FPGA
- Domain-specific hybrid FPGA architecture
  - floating point applications
  - improve slices usage by *18 times* over XC2V3000
- Optimized interface for floating point hybrid FPGA
- Internal architecture of coarse-grained blocks: little study

2. Contributions

- Methodology
  - optimize the floating point hybrid FPGA
  - consider mixture of FPUs
- Study of FPUs architecture
  - use common subgraph extraction
- Quantitative system level analysis
  - speed: ***16.7%*** faster with only floating point add/mult
  - area: ***27.4%*** smaller with highest density system
  - routing resources: ***14.8%*** fewer with highest density system

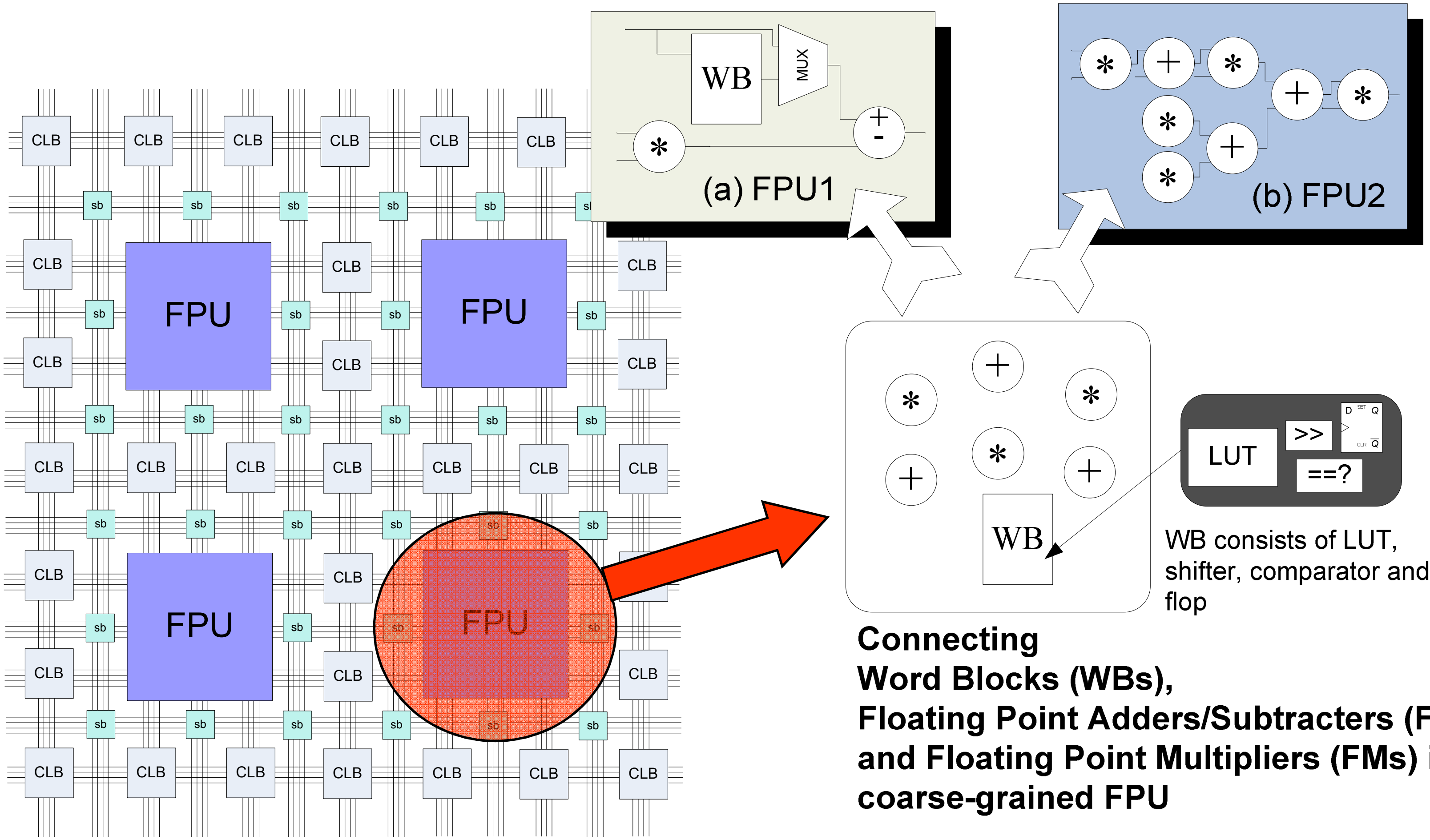
3. Methodology: Fine-grained assumption





Fine-grained (configurable logic block)

Coarse-grained assumption

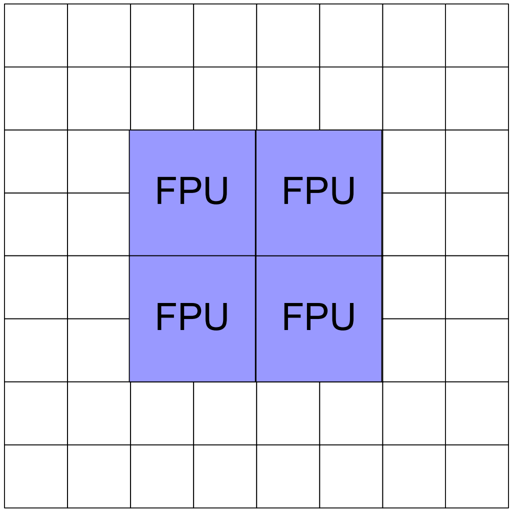


WB consists of LUT, shifter, comparator and flip flop

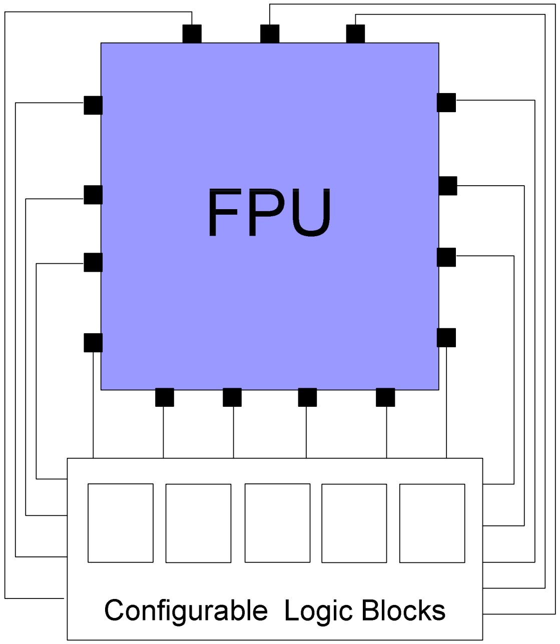
Connecting
Word Blocks (WBs),
Floating Point Adders/Subtracters (FAs)
and Floating Point Multipliers (FMs) into
coarse-grained FPU

Interface assumption

- FPUs are square
- FPUs are in the center of FPGA
- FPU pins on four sides



FPUs are in the center of FPGA



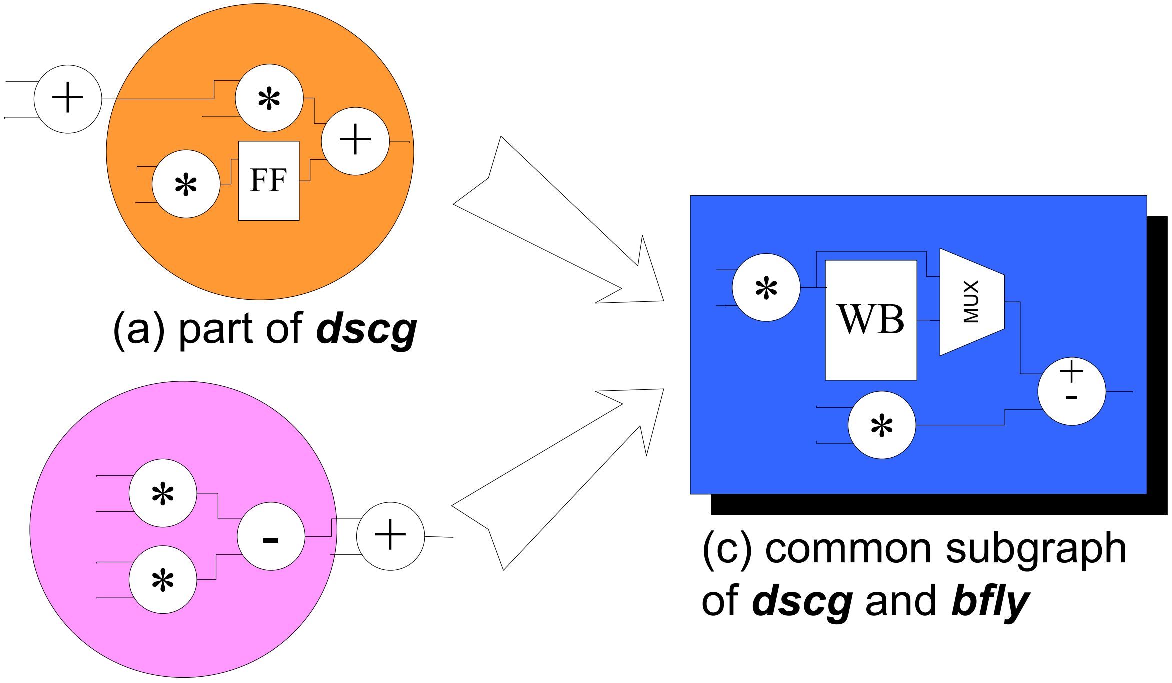
CLBs connect to four sides of FPU

Optimization parameters

- **Internal optimization of FPU**
  - performance of individual FPU
  - according to different combination of Word Blocks, Floating Point Adders/Subtracters and Multipliers

- **System level optimization**
  - density and flexibility of FPU
  - affect area, speed and routing resources of system

4. Study of FPUs architecture: Common subgraph extraction



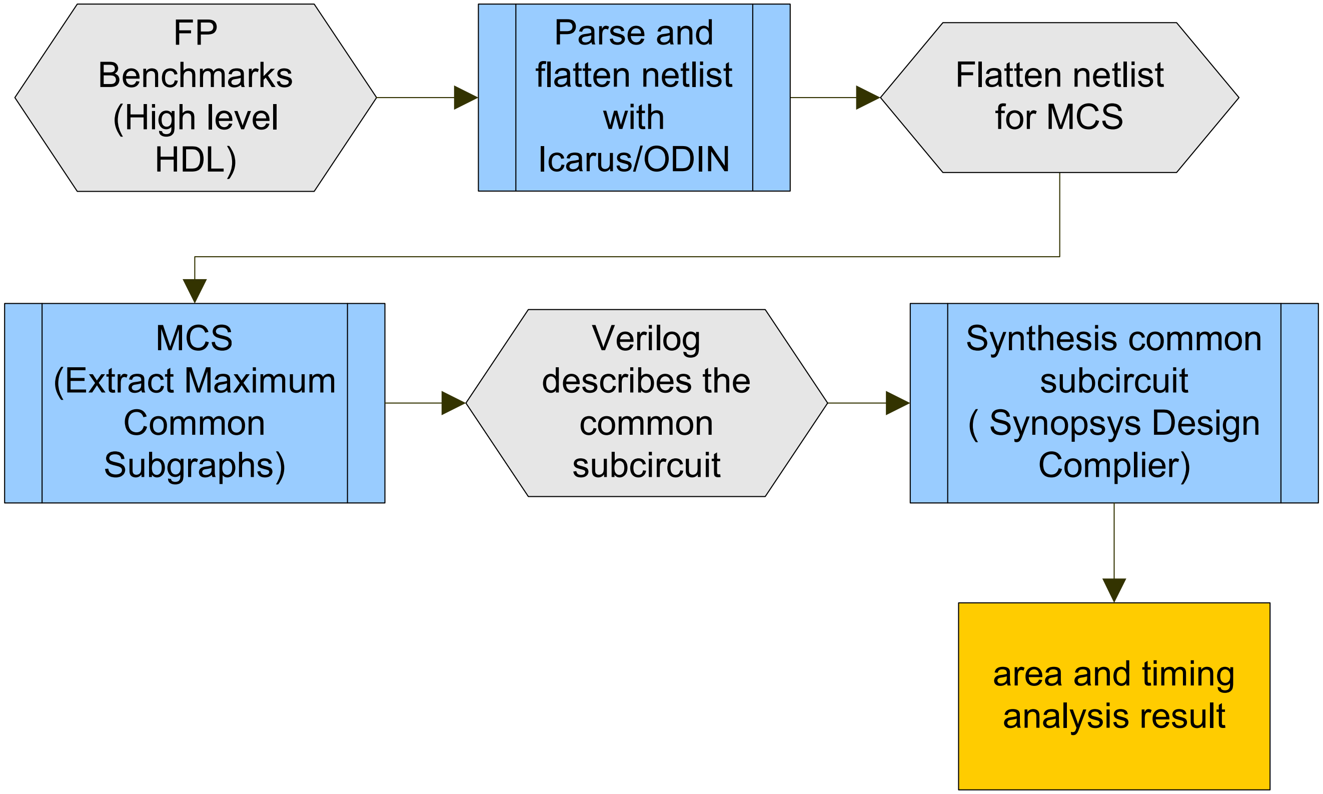
(a) part of ***dscg***

(b) part of ***bfly***

(c) common subgraph of ***dscg*** and ***bfly***

An example of common subgraph extraction for floating point computation

Common subgraph extraction


FP Benchmarks (High level HDL)
Parse and flatten netlist with Icarus/ODIN
Flatten netlist for MCS
MCS (Extract Maximum Common Subgraphs)
Verilog describes the common subcircuit
Synthesis common subcircuit (Synopsys Design Complier)
area and timing analysis result

Design flow of common subgraph extraction

Benchmarks

- dscg – digital sine cosine generator
- bfly – basic component of Fast Fourier Transform
- fir4 – 4-tap finite impulse response filter
- ode – ordinary differential equation
- mm3 – 3x3 matrix multiplication circuit
- bgm – Monte Carlo simulations of interest rate model derivatives
- syn2 and syn7 – synthetic benchmark circuits

Benchmarks	dscg	bfly	fir4	ode	mm3	bgm	syn2	syn7
No. of FAs	4	4	3	3	2	9	5	25
No. of FMs	4	4	4	2	3	11	4	25

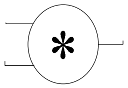
Number of FAs and FMs used in each benchmarks

Results

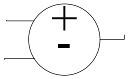
- 41 common subgraphs
- Different area, speed and flexibility
- 3 systems to explore
  - purely FA/FM FPGA
  - highest flexibility FPGA
  - highest density FPGA

Selection of systems – Purely FA/FM FPGA

- ***Purely FA/FM FPGA*** - 2 types of FPU



25x FM



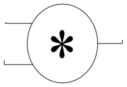
25x FA

Graph no.	Occurrence in benchmarks								Area in CLB	Delay (ns)
	dscg	bfly	fir4	ode	mm3	bgm	syn2	syn7		
FA	4	4	3	3	2	9	5	25	7	2.77
FM	4	4	4	2	3	11	4	25	14	3.18

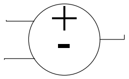
Statistic of the occurrence of FA and FM in benchmarks

Selection of systems – Highest flexibility FPGA

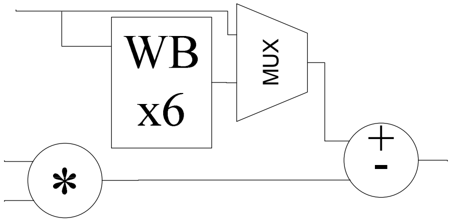
- **Highest flexibility FPGA:** ***FPGA_12_15_26*** - 5 types of FPU



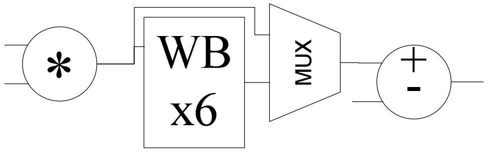
7x FM: *14 CLBs, 3.18ns*



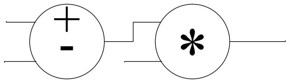
7x FA: *7 CLBs, 2.77ns*



16x Graph 12: *38 CLBs, 3.66ns*



2x Graph 15: *38 CLBs, 3.53ns*



2x Graph 26: *20 CLBs, 3.39ns*

The FPUs are most common in the benchmarks

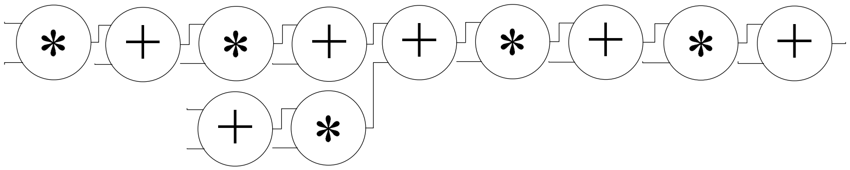
Selection of systems – Highest flexibility FPGA

Graph no.	Number of FA, FM and WB represented in benchmarks								Total number of FA, FM and WB represented
	dscg	bfly	fir4	ode	mm3	bgm	syn2	syn7	
12	16	4	6	8	2	44	0	32	112
15	16	4	6	2	10	30	0	32	100
26	4	0	0	2	0	10	2	30	48

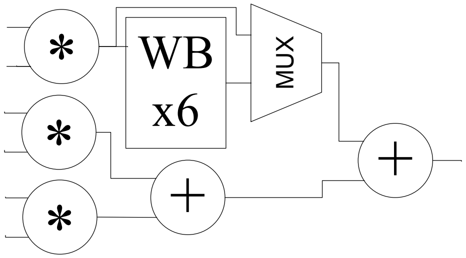
Order of the subgraphs to be selected by total number of FA, FM and WB represented in benchmarks

Selection of systems – Highest density FPGA

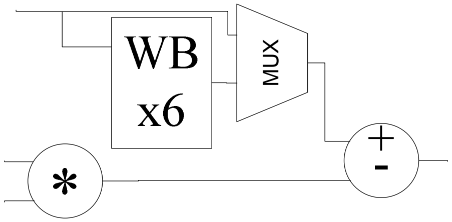
- **Highest density FPGA:** ***FPGA_41_20_37_12_26*** - 7 types of FPU



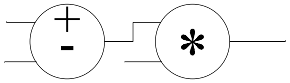
1x Graph 41: *97 CLBs, 3.68ns*



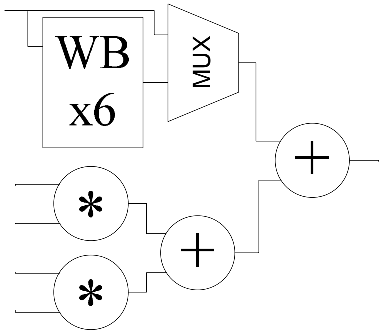
1x Graph 20: *66 CLBs, 3.54ns*



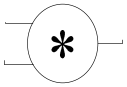
8x Graph 12: *38 CLBs, 3.66ns*



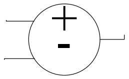
3x Graph 26: *20 CLBs, 3.39ns*



2x Graph 37: *57 CLBs, 3.74ns*



5x FM: *14 CLBs, 3.18ns*



4x FA: *7 CLBs, 2.77ns*

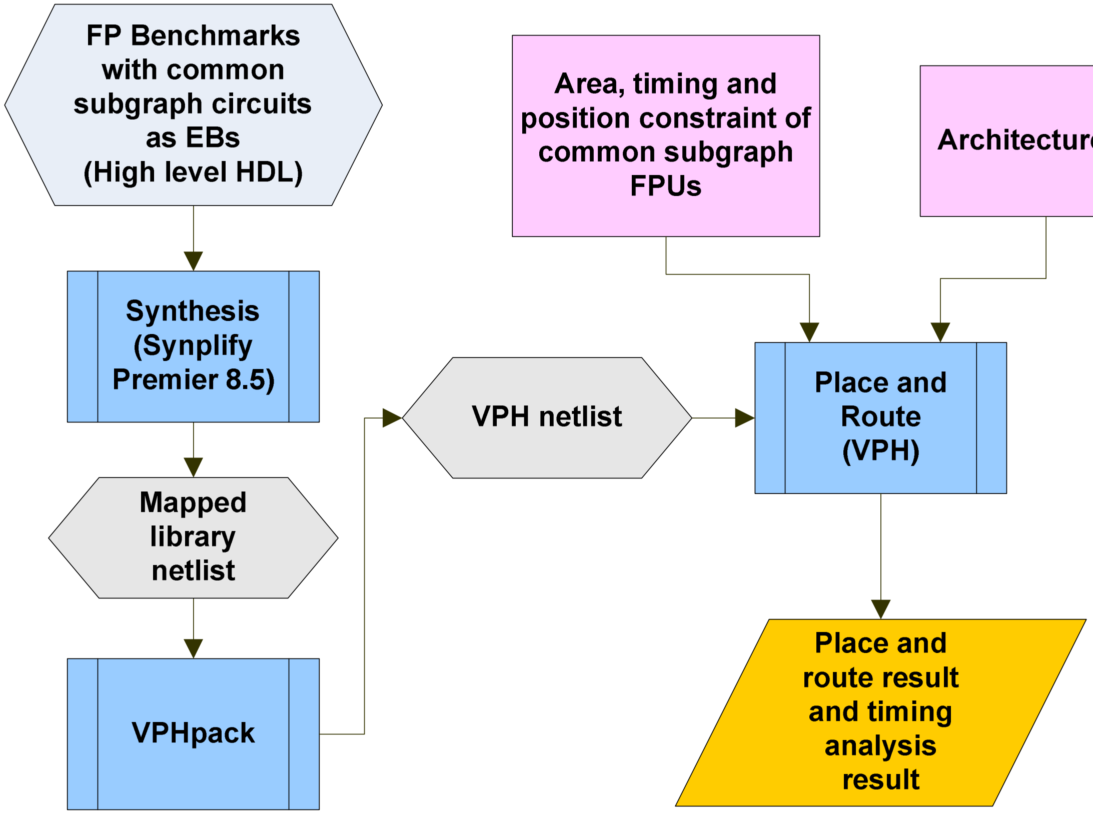
The FPUs have highest area efficiency in the benchmarks

Selection of systems – Highest density FPGA

Graph no.	No. of FA in the subgraph	No. of FM in the subgraph	Total number of FA, FM (no. of WB)
41	6	5	11
20	2	3	5 (6)
37	2	2	4(6)
12	1	1	2 (6)
26	1	1	2

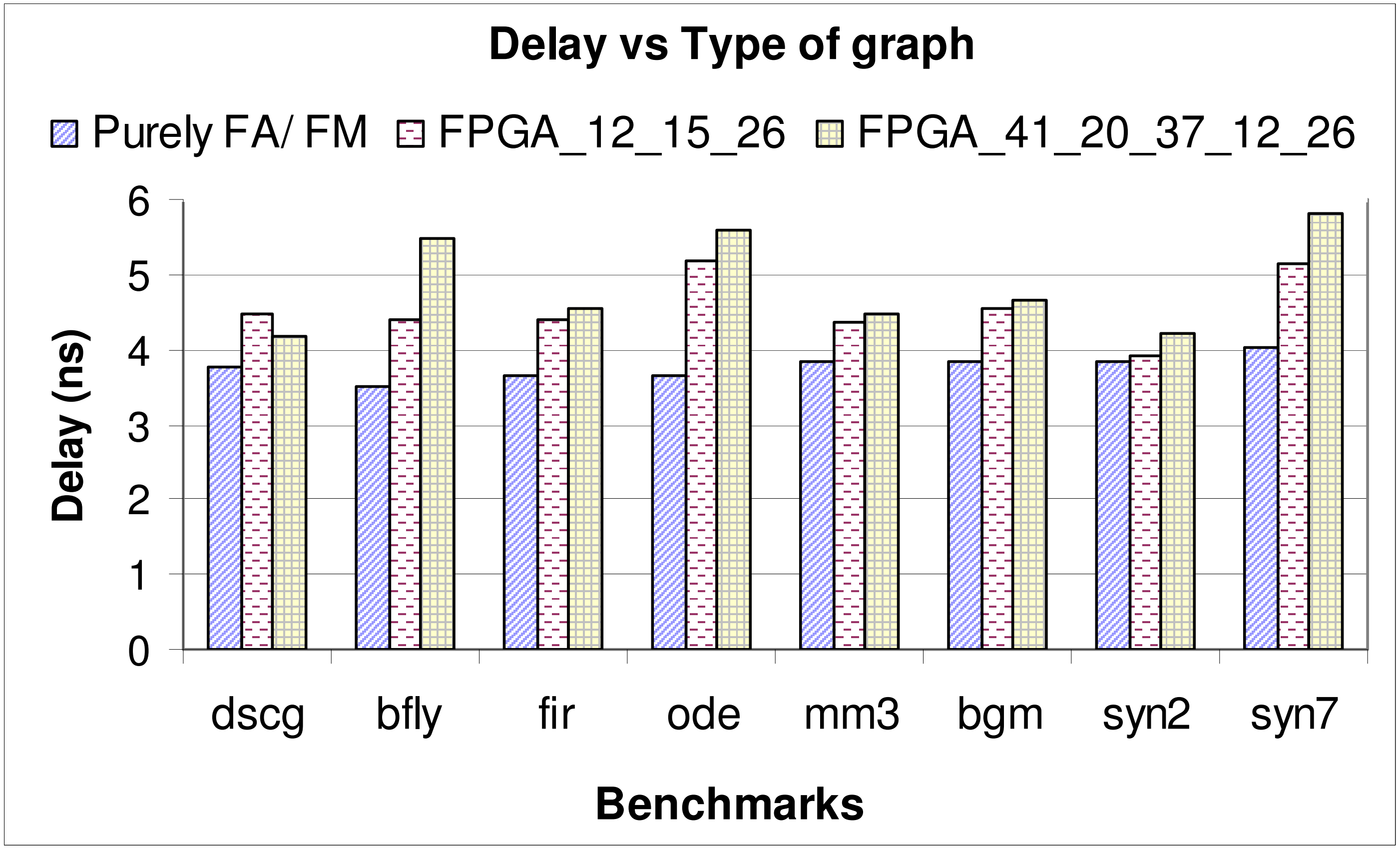
Order of the subgraphs to be selected by area efficiency of each subgraph (area of WB is insignificant compared to FA and FM)

5. Quantitative system level analysis Versatile Place and Route for Hybrid FPGAs (VPH)


FP Benchmarks with common subgraph circuits as EBs (High level HDL)
Synthesis (Synplify Premier 8.5)
Mapped library netlist
VPHpack
VPH netlist
Area, timing and position constraint of common subgraph FPUs
Architecture file
Place and Route (VPH)
Place and route result and timing analysis result

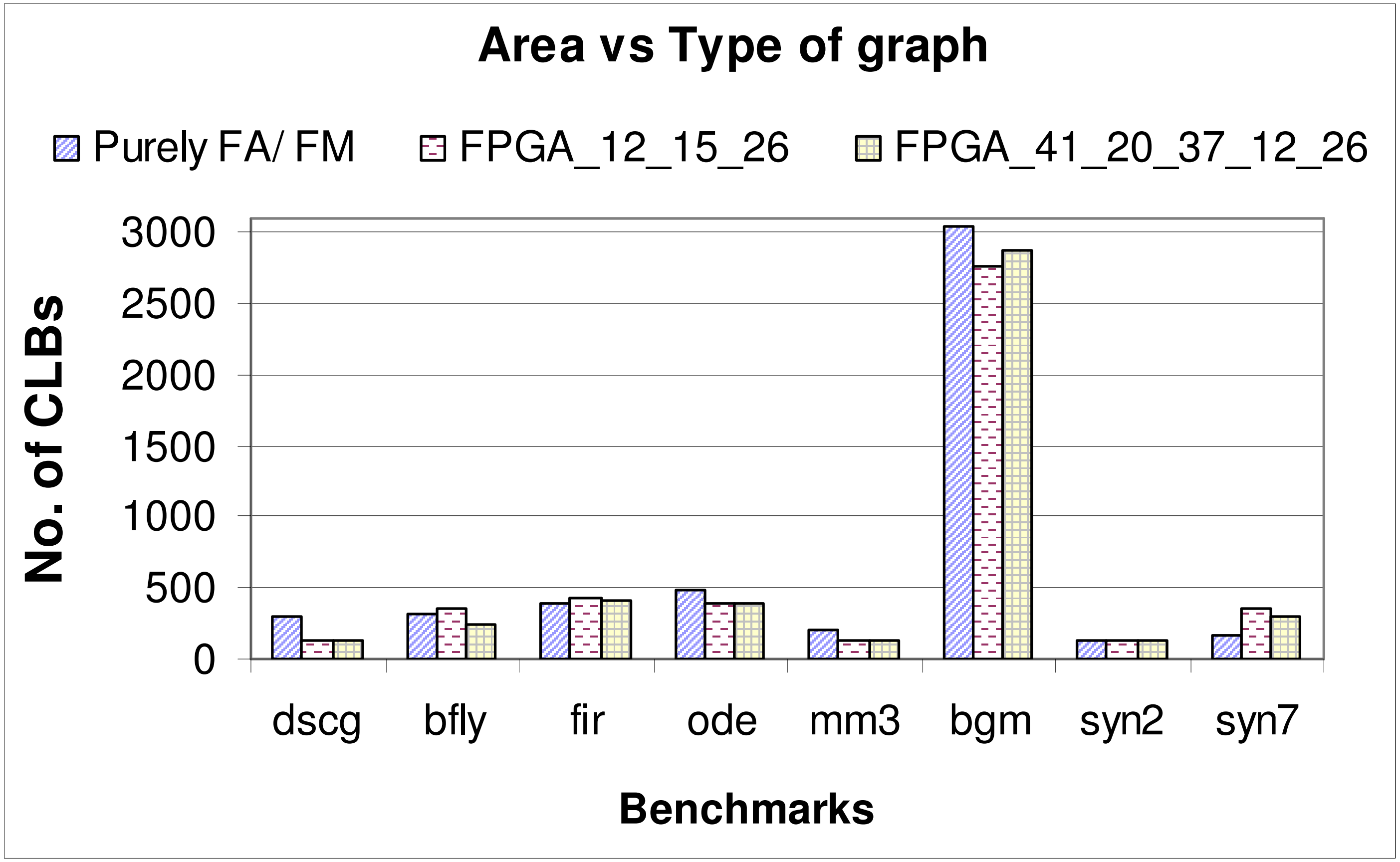
Design flow of VPH with common subgraph

Delay impact



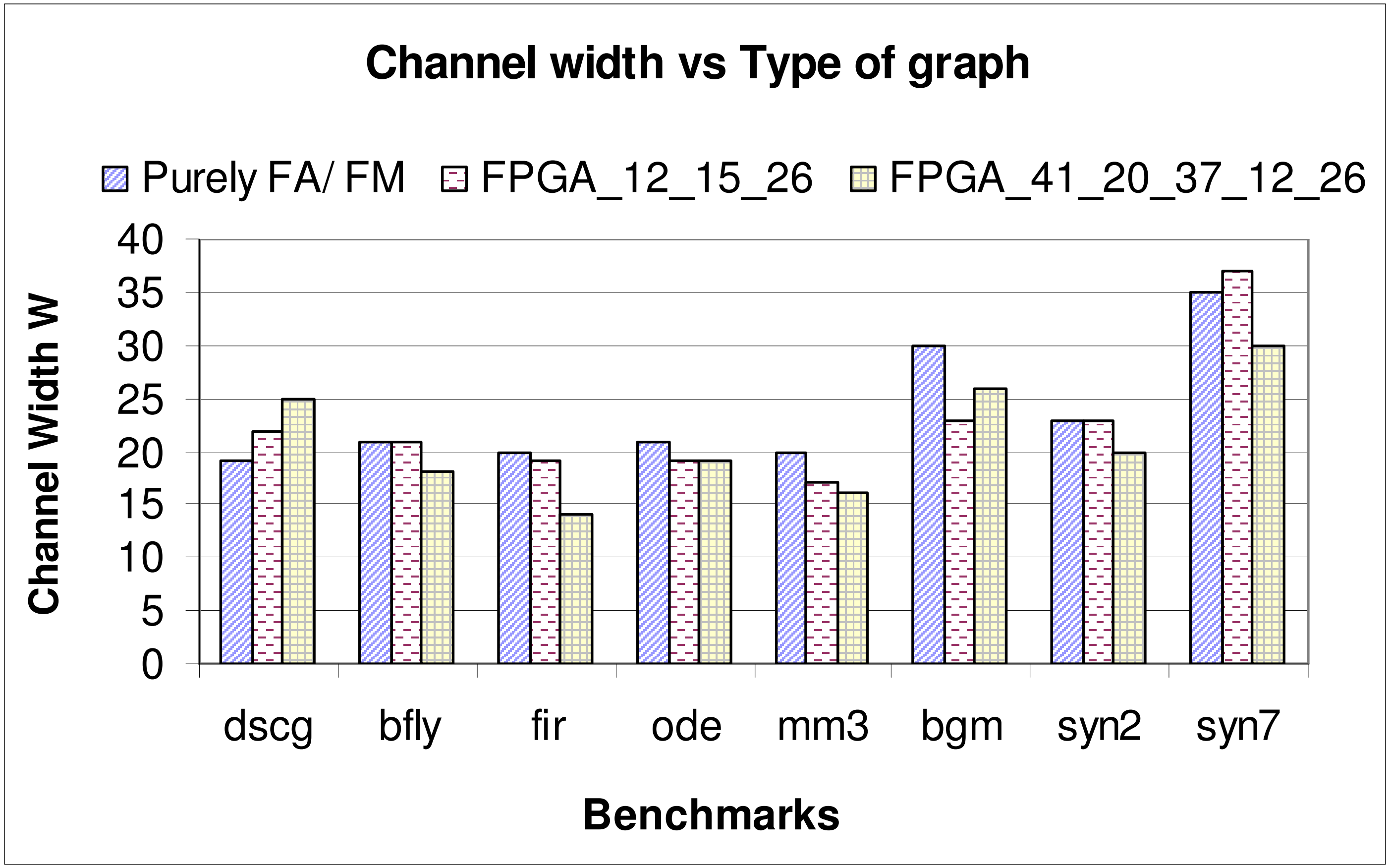
Purely FA/FM is at least 16.7% faster than *FPGA_12_15_26*

Area impact



FPGA_41_20_37_12_26 reduces 27.4% area compared to *Purely FA/FM*

Routing resources impact



FPGA_41_20_37_12_26 requires 14.8% fewer routing resources than *Purely FA/FM*

6. Conclusion

- Common subgraph extraction
  - determine optimized floating point coarse-grained blocks in hybrid FPGAs

- Optimizing floating point hybrid FPGA
  - FAs and FMs only system has highest speed, ***16.7%*** faster
  - higher density subgraphs reduce ***27.4%*** of system area
  - higher density subgraphs reduce ***14.8%*** of routing resources

- Future: apply approach to FPGAs for different application domains
  - medical imaging
  - financial analysis