UiT

THE ARCTIC
UNIVERSITY
OF NORWAY

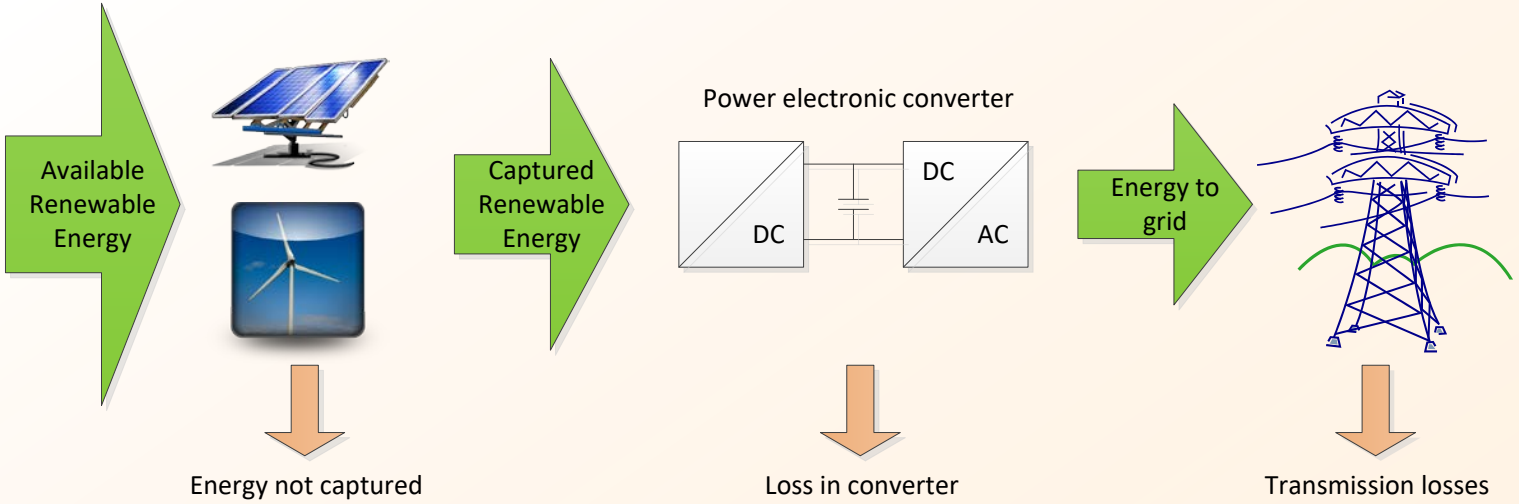
Advanced Model Predictive Control Algorithm for Inverters as a Low-cost Solution in ZynQ

Bjarte Hoff
PhD Candidate

Outline

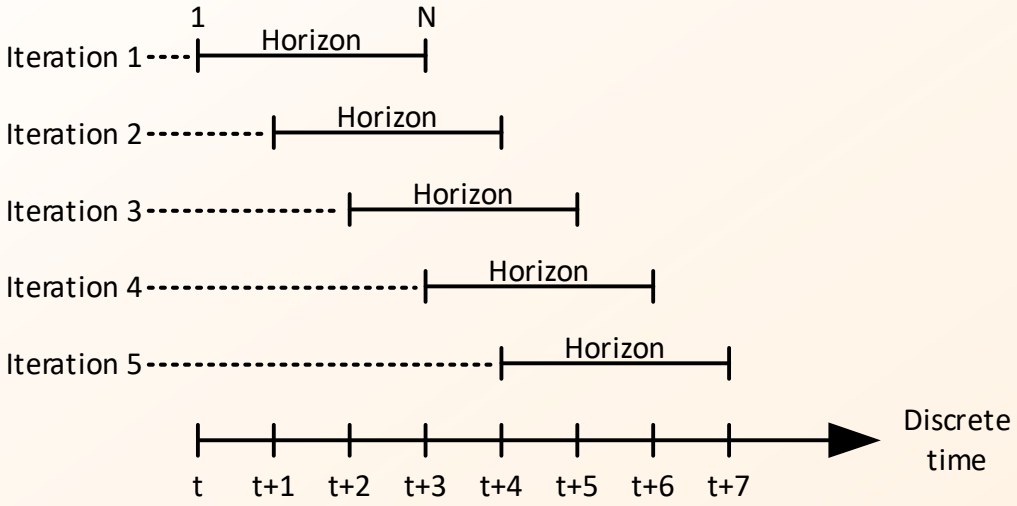
- Motivation and application
- System overview
- Cascaded model predictive control (MPC)
- Hardware setup
- Inner control loop implementation
- Outer control loop implementation
- Performance
- Conclusion

Motivation and application



- Renewable energy is connected to the grid using power electronics
- Performance of the energy conversion depends on the control algorithm

Introduction to MPC



Finite control set MPC

Finite number of solutions:

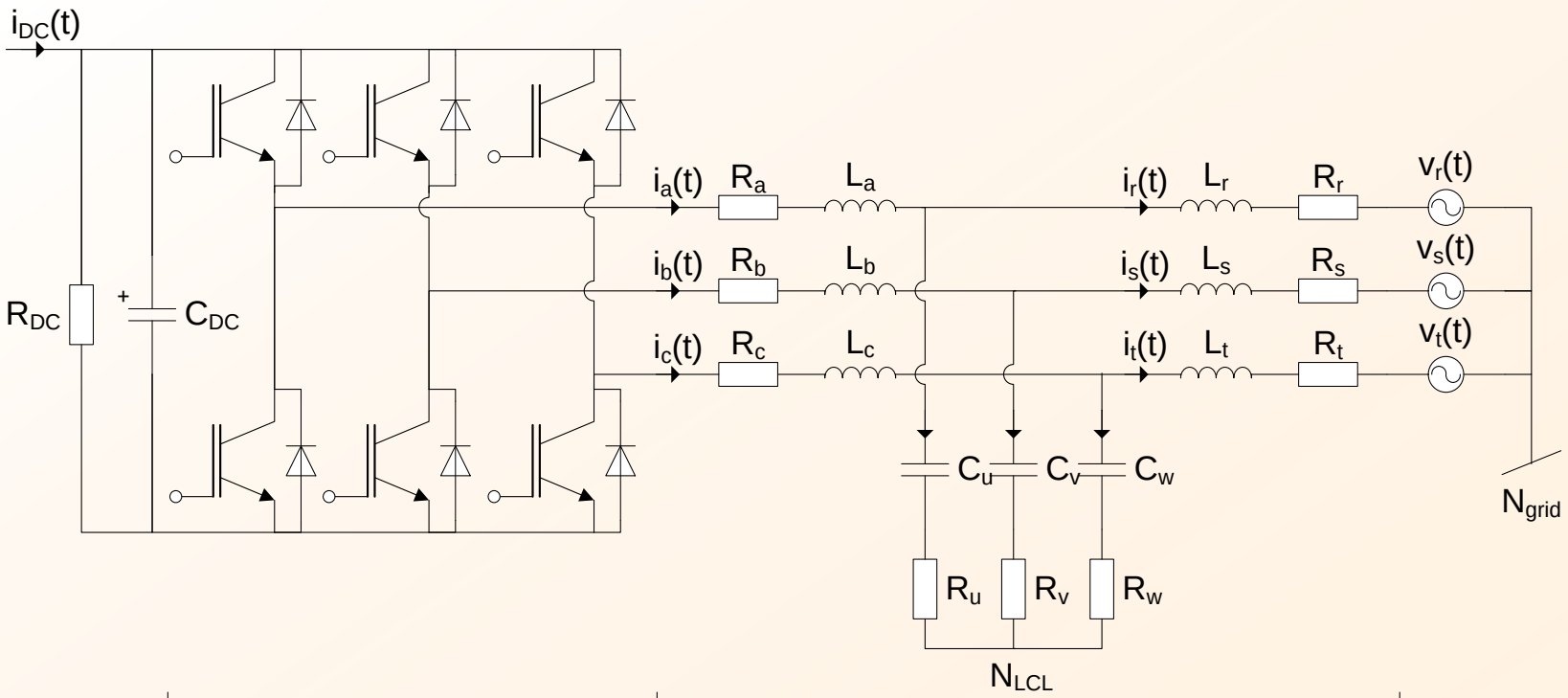
$$\min_{\mathbf{u}\in\{0,1\}} \quad \left\| \mathbf{y}^* - \mathbf{y}_{k+1} \right\|_2^2$$
$$s.t. \quad \mathbf{y}_{k+1} = G(\mathbf{u}),$$

Continuous control set MPC

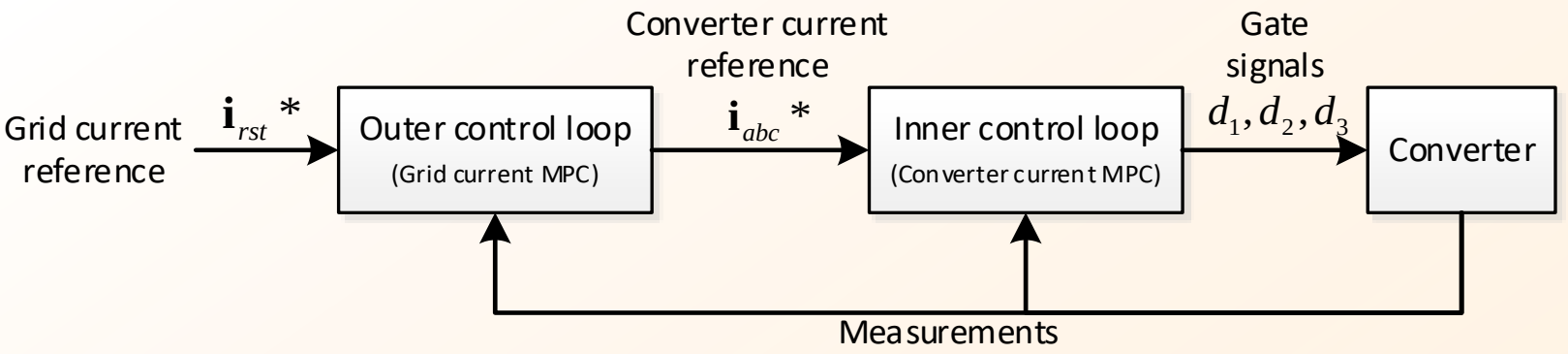
Infinite number of solutions:

$$\min_{x} \quad q(x) = \frac{1}{2} x^T G x + x^T c$$
$$s.t. \quad a_i^T x = b_i, \qquad i \in \mathcal{E}$$
$$a_i^T x \geq b_i, \qquad i \in \mathcal{I}$$

Two-level three-phase converter



Cascaded MPC



Outer control loop (CCS-MPC)

$$\min_{\substack{\Delta \mathbf{u} \\ \to k-1}} \quad \frac{1}{2} \Delta \mathbf{u}_{\to k-1}^{T} H^T H \Delta \mathbf{u}_{\to k-1} + \Delta \mathbf{u}_{\to k-1}^{T} g$$

$$s.t. \quad \Delta \mathbf{u}_{lb} \le \Delta \mathbf{u}_{\to k-1} \le \Delta \mathbf{u}_{ub},$$

$$\mathbf{x}_{lb} \le \mathbf{x}_{\to k} \le \mathbf{x}_{ub},$$

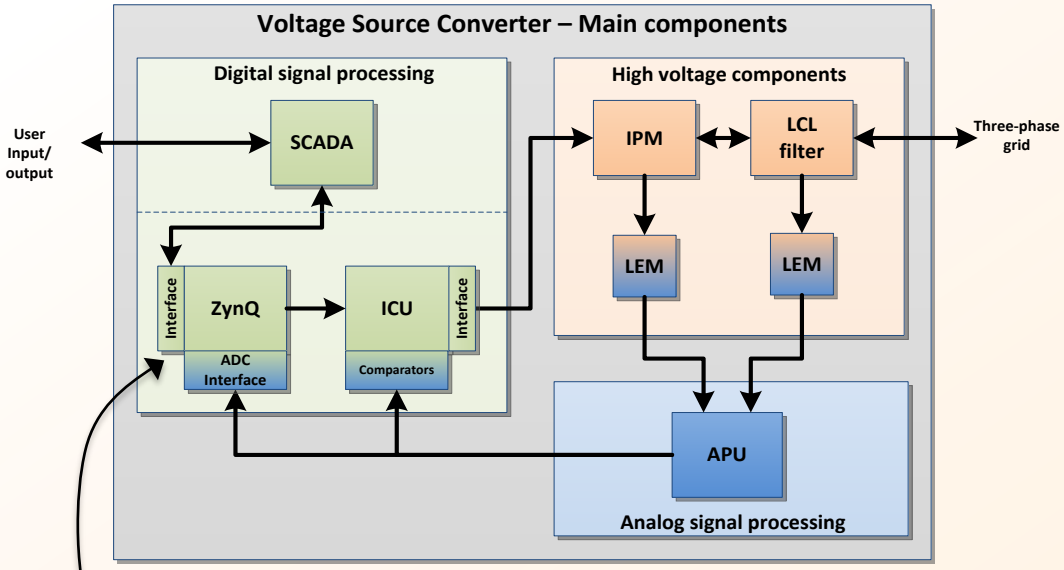
Inner control loop (FCS-MPC)

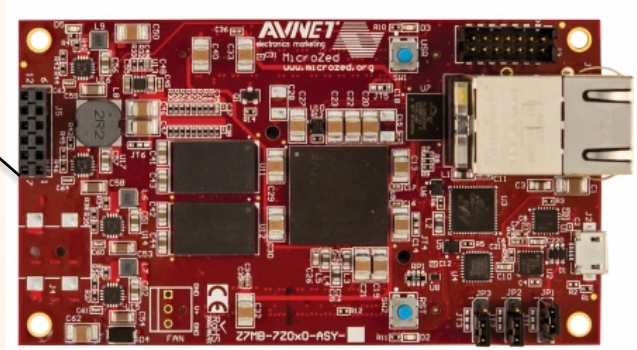
$$\min_{d_1, d_2, d_3} \quad \left\| \mathbf{i}_{abc}^* - \mathbf{y}_{k+1} \right\|_2^2$$

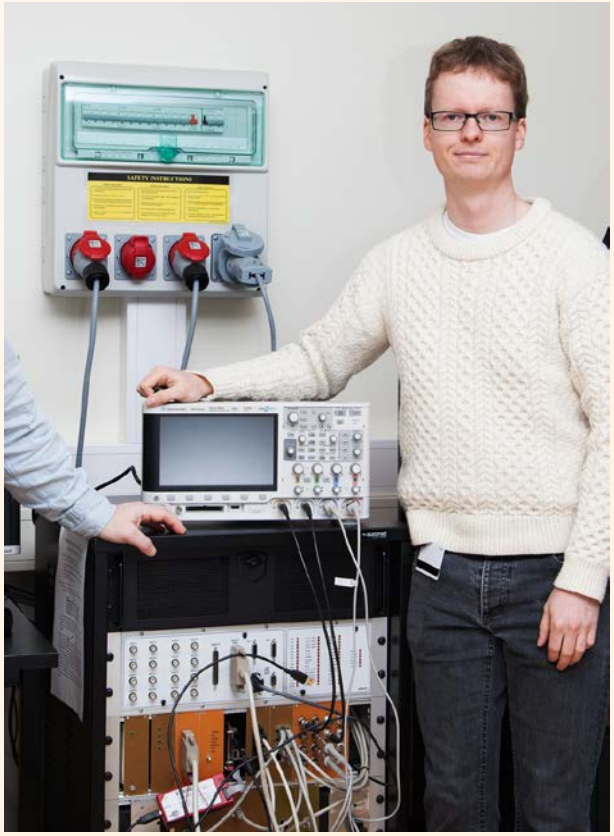
$$s.t. \quad \mathbf{x}_{k+1} = A(d_1, d_2, d_3)\mathbf{x}_k$$

$$\mathbf{y}_{k+1} = C\mathbf{x}_{k+1},$$

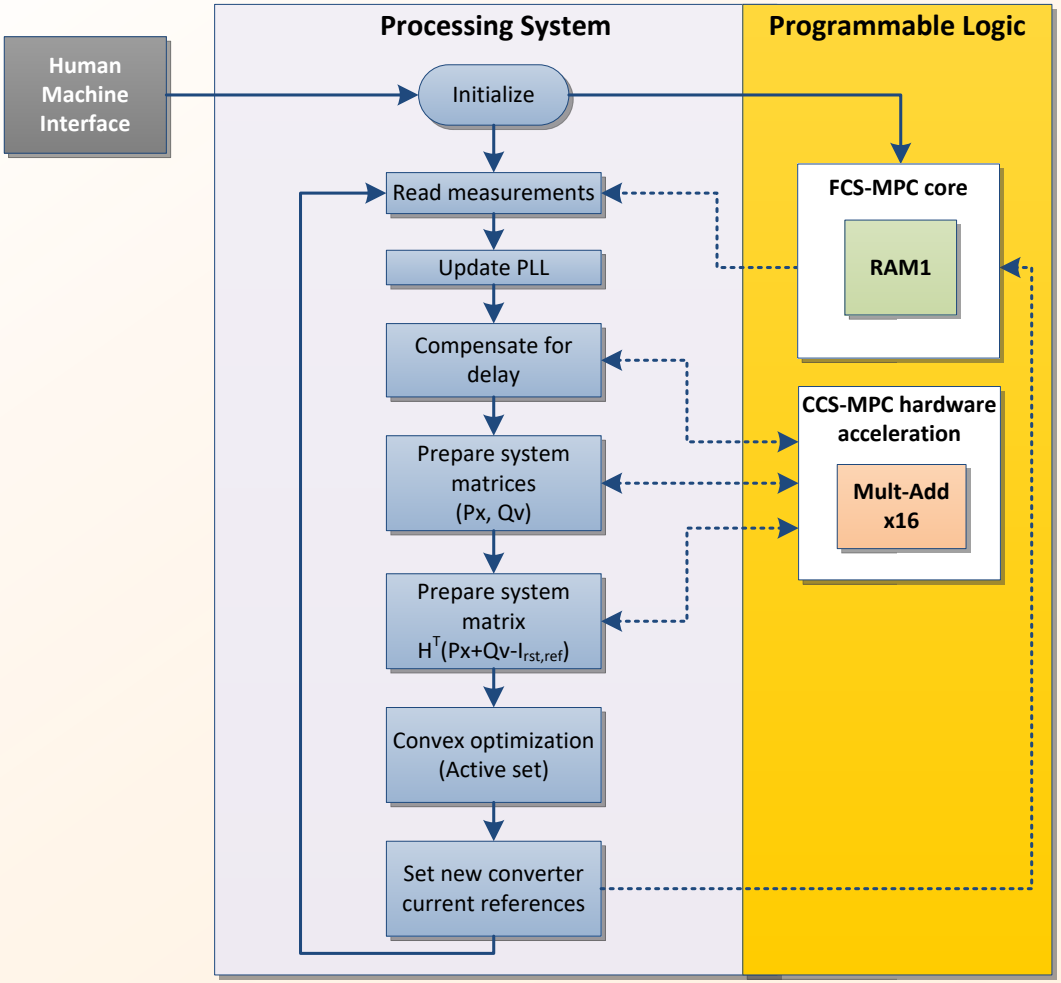
Hardware setup



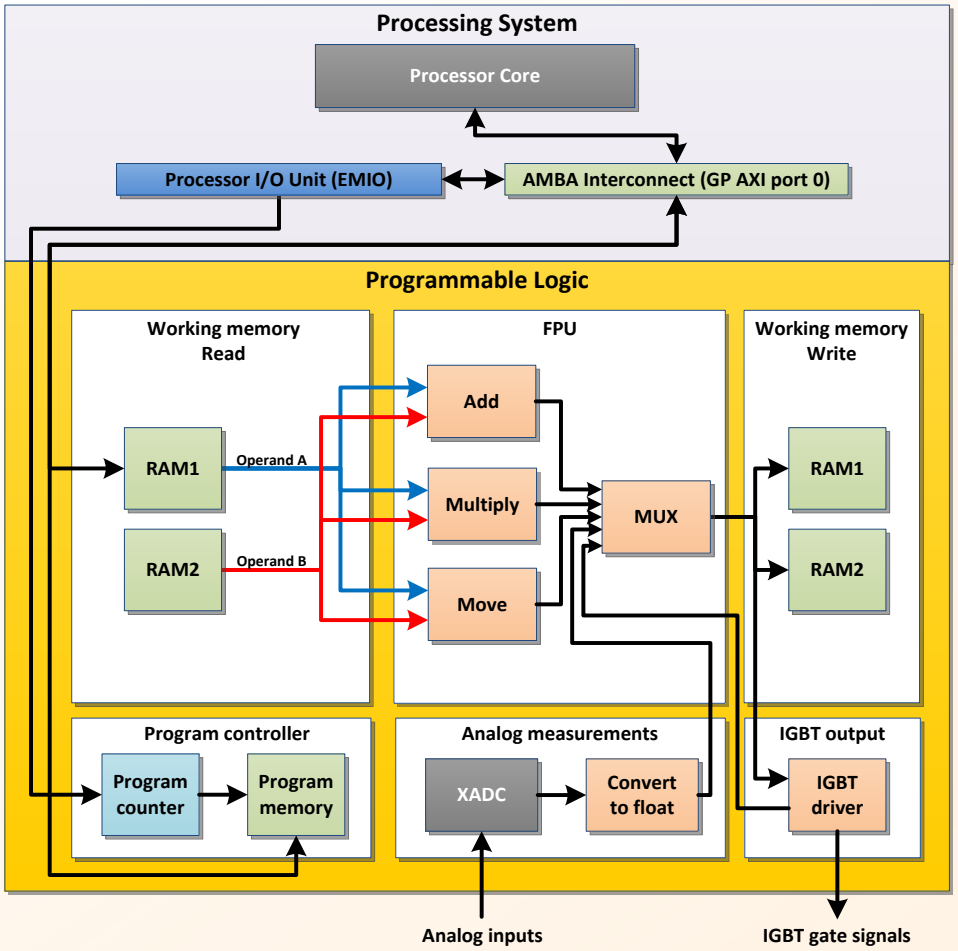


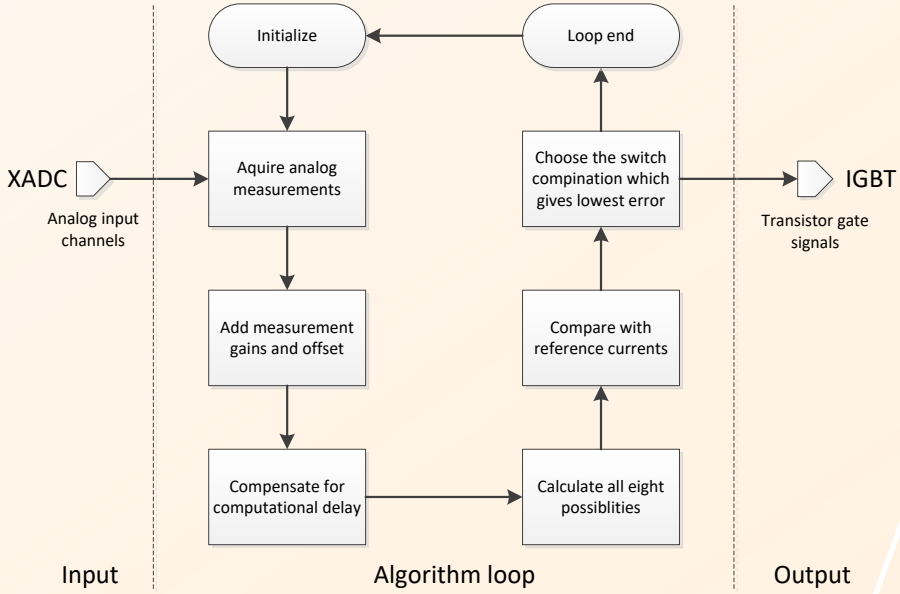


Cascaded MPC control loop

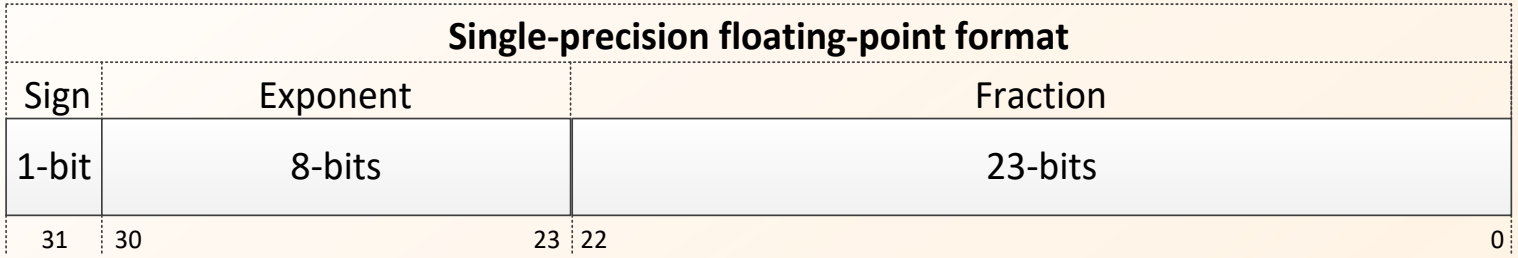


FCS-MPC hardware acceleration





Single-precision floating-point format



$$decimal\ value = (-1)^{sign}\, 2^{\exp onent - 127}\,(1 \cdot\ fraction)$$

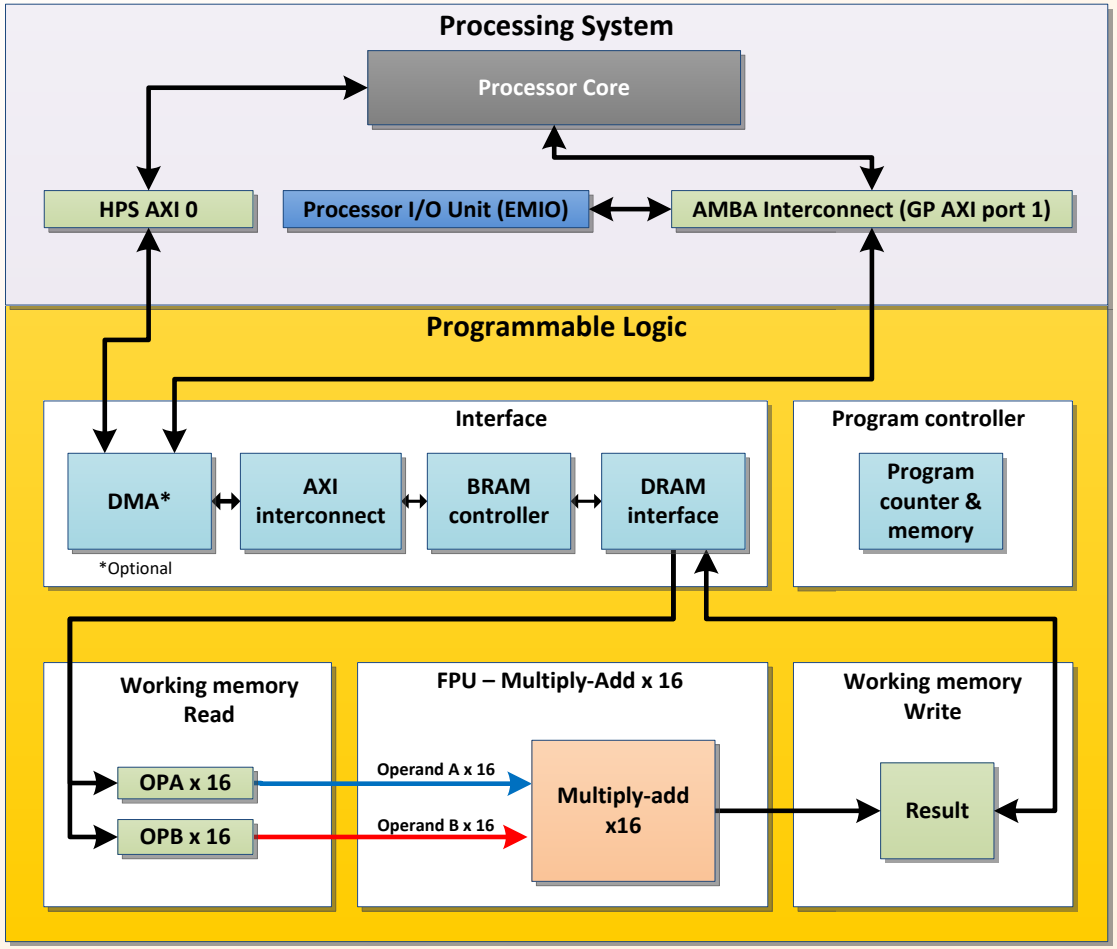
- Seven different values are compared by decomposing the floating point number into bits

FCS-MPC Firmware

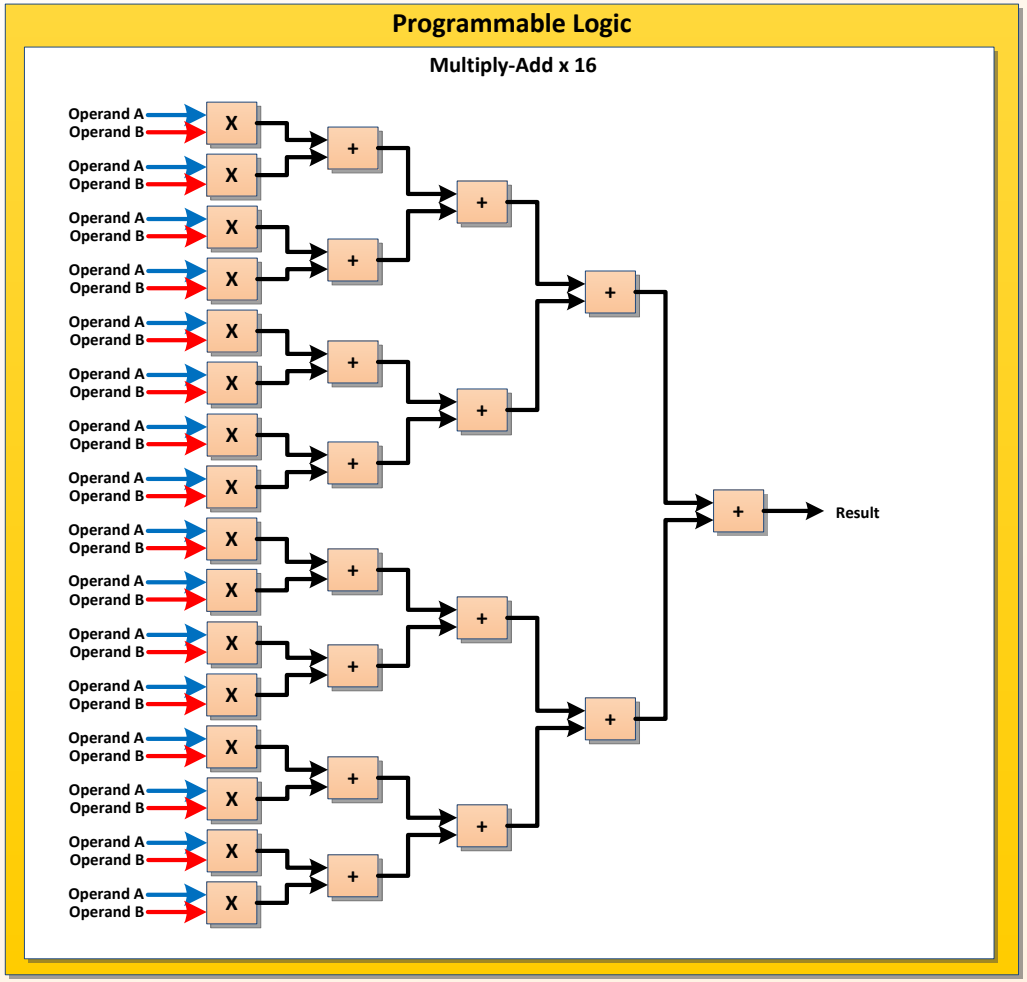
PC	Addr_RAM1B	Addr_RAM2A	Addr_RAM2B	RAM1B_W	RAM2A_W	EN_mult	EN_add	EN_comp	EN_i2f	R_Data_B(2)	R_Data_A(1)	R_comp	R_i2f	R_mult	R_add	ADC_convst	Load_PC	D_out	Instruction
0	31		1			1			1										00111110000000001001001000000000
1	32		2			1			1										01000000000000010001001000000000
2	33		3			1			1										01000010000000011001001000000000
3	34		4			1			1										01000100000000100001001000000000
4	35		5			1			1										01000110000000101001001000000000
5	36		6			1			1										01001000000000111000100100000000
6	37		7			1			1										01001010000000111001001000000000
7	38		8			1			1										01001100000001000001001000000000
8	39		9			1			1										01001110000001001001001000000000
9	40	1	10		1	1			1					1					01010000000101010011001000010000
10	21	2	1		1	1	1		1					1					00101010001000001011101000010000
11	22	3	2		1	1	1		1					1					00101100001100010011101000010000
12	23	4	3		1	1	1		1					1					00101110010000011011101000010000
13	24	5	4		1	1	1		1					1					00110000010100100011101000010000
14	25	6	5		1	1	1		1					1					00110010011000101011101000010000
15	26	7	6		1	1	1		1					1					00110100011100110011101000010000
16	27	8	7		1	1	1		1					1					00110110100000111011101000010000
17	28	9	8		1	1	1		1					1					00111000100101000011101000010000
18	29	10	9		1	1	1		1					1					00111010101001001011101000010000
19	30		10				1		1										00111100000001010000101000000000
20							1		1										00000000000000000000101000000000
21							1		1										00000000000000000000101000000000
22	11	1		1	1		1		1						1				00010110000100000110101000001000
23	12	2		1	1		1		1						1				00011000001000000110101000001000
24	13	3		1	1		1		1						1				00011010001100000110101000001000
25	14	4		1	1		1		1						1				00011100010000000110101000001000
26	15	5		1	1		1		1						1				00011110010100000110101000001000
27	16	6		1	1		1		1						1				00100000011000000110101000001000
28	17	7		1	1		1		1						1				00100010011100000110101000001000
29	18	8		1	1		1		1						1				00100100100000000110101000001000
30	19	9		1	1		1		1						1				00100110100100000110101000001000
31	20	10		1	1		1		1						1				00101000101000000110101000001000

COE-file to program the Firmware is generated from Excel using a macro

CCS-MPC hardware acceleration



Multiply-Add x16



DRAM Interface

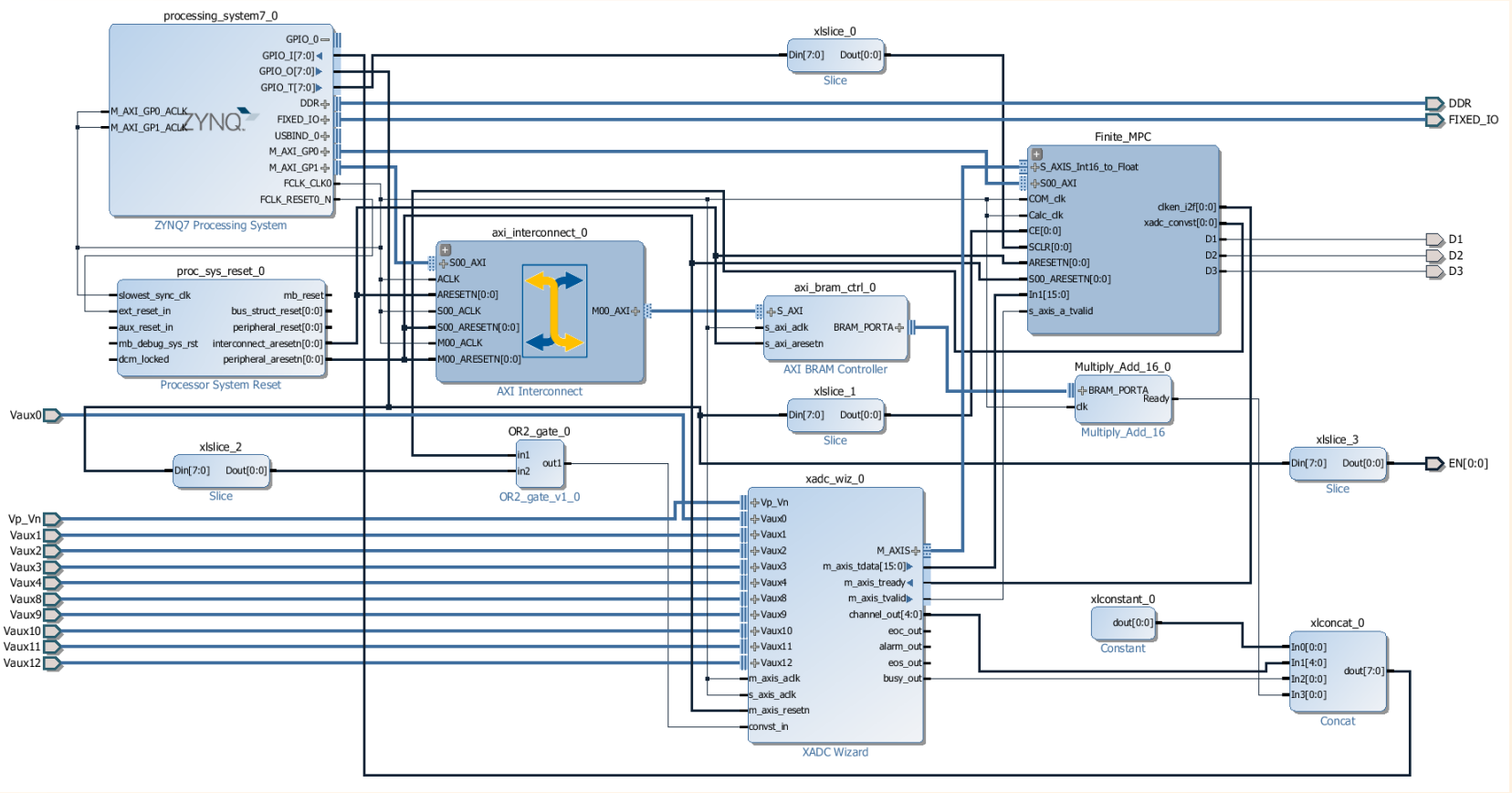
```
 -- Output BRAM clock
    DRAM_clk_out <= clka;

    -- Output address
    DRAM_addr_in <= addra(12 downto 6);
    DRAM_addr_out <= addra(8 downto 2);

    -- RAM selected for write operations
    with (addra(13) & addra(5 downto 2)) select RAM_select <=
    "00000000000000000000000000000001" when "00000",
    "00000000000000000000000000000010" when "00001",
    "00000000000000000000000000000100" when "00010",
    "00000000000000000000000000001000" when "00011",
    "00000000000000000000000000010000" when "00100",
    "00000000000000000000000000100000" when "00101",
```

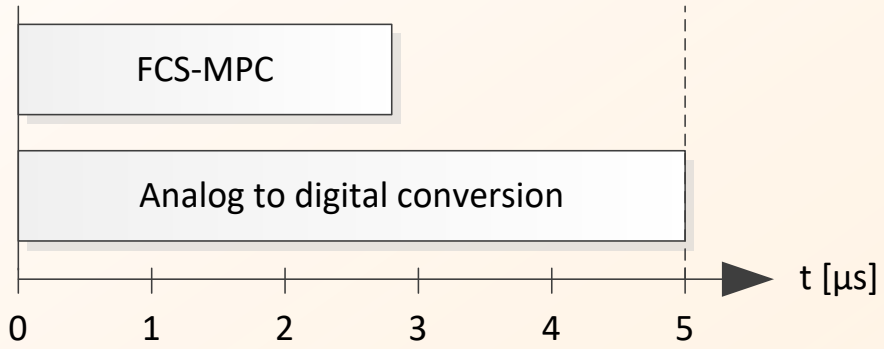
etc…

Xilinx Vivado – Block schematic screenshot

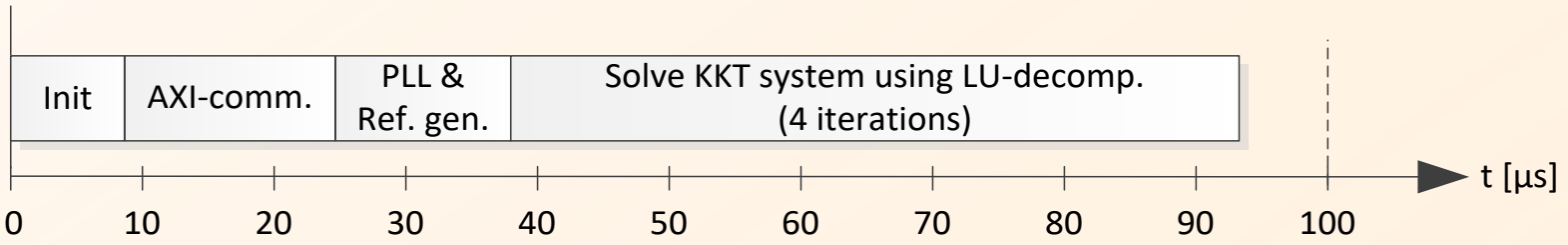


Computational performance

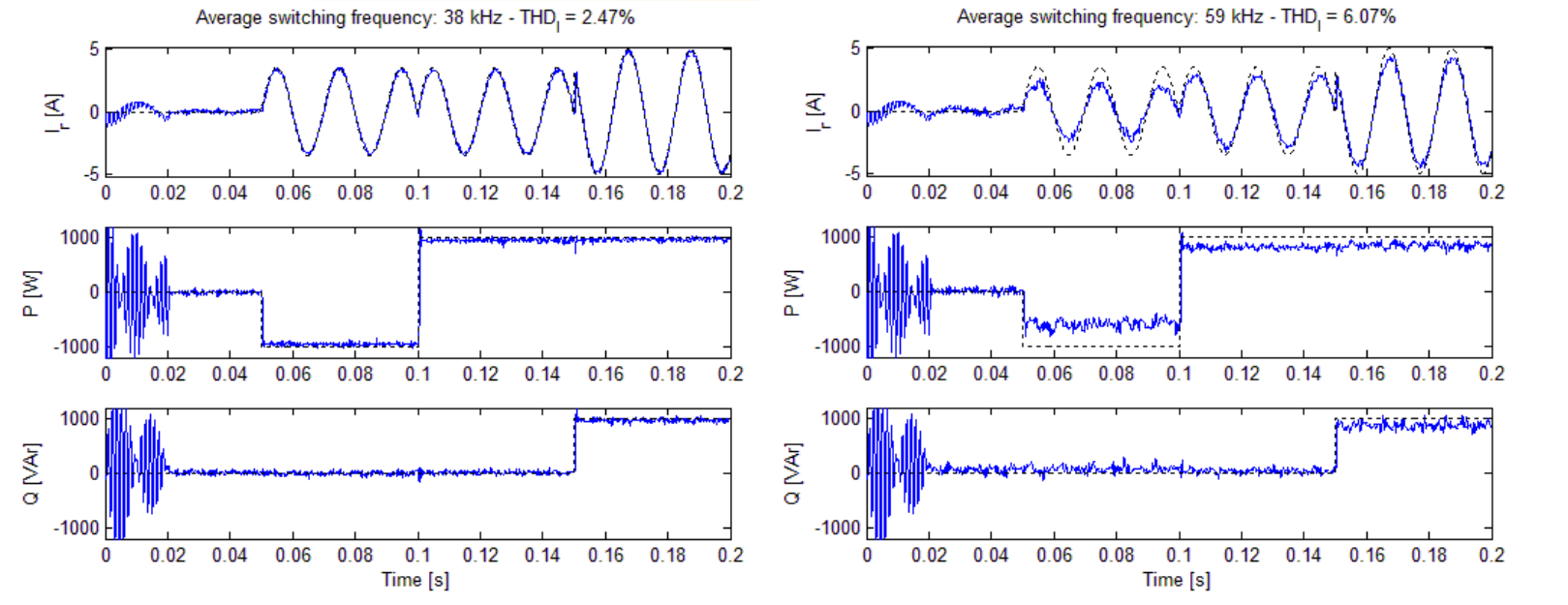
- Inner control loop (FCS-MPC):



- Outer control loop (CCS-MPC):



Simulation results



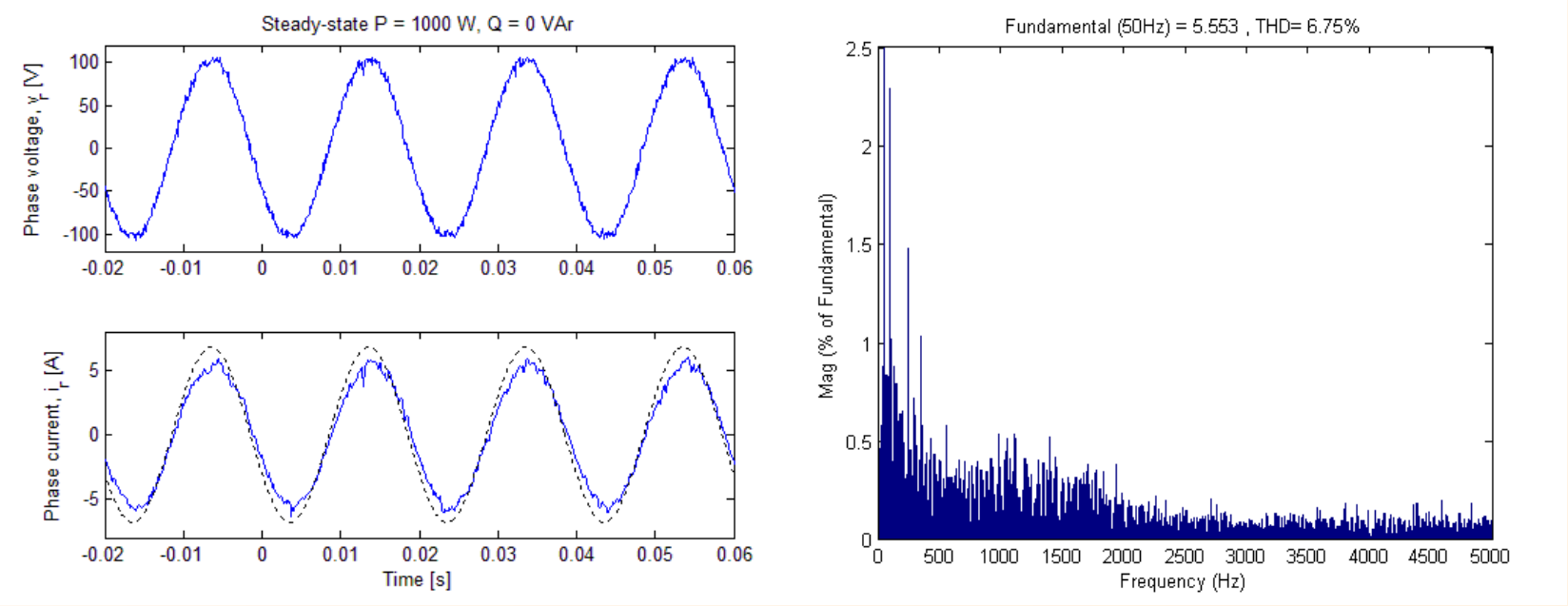
Dead-time: 1µs

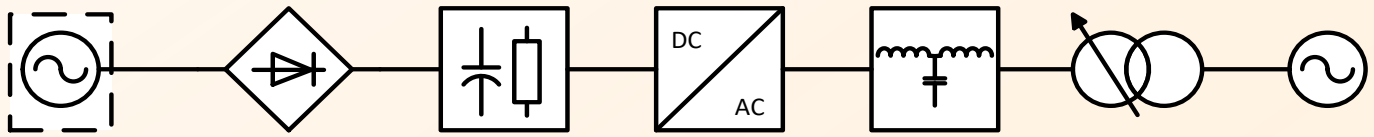
SiC-based converter

Dead-time: 5µs

IGBT-based converter

Experimental results





Programmable three-phase AC source	Three-phase rectifier	DC filter capacitor and load	Converter	LCL filter	Variac	Grid 400V, 50Hz

Conclusion

- Cascaded MPC has been successfully implemented in hardware
- Performance requirements are met by taking the advantages of both microprocessor and FPGA
- The algorithm is implemented as a low-cost solution in Xilinx ZynQ
- Improved experimental performance is expected when applied to a modern converter with fast low-loss transistors