LabC DFT report

r11943043 潘奕亘
r11943012 曾維雋
b07901073 吳秉軒

A. Introduction

In LabC basic topic 1, we are going to design a 1024 points DFT (Discrete Fourier Transform) circuit, transforms a sequence of N complex numbers into another sequence of complex numbers, which is defined by

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$
$$= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right)\right]$$

Our target is to maximize the score $\frac{T_{clock} \, x \, F_{max}}{\tau_{simulation}}$, so we need to properly utilize the hardware resource and maximize throughput.

B. Implementation

1. Baseline

```
void dft(DTYPE real_sample[1024], DTYPE imag_sample[1024],DTYPE real_op[1024],DTYPE imag_op[1024])
{
    int k, n;
    int index;

    for (k = 0; k < 1024; k++)
    {
        for (n = 0; n < 1024; n++){
            index = k * n & 1023;
            real_op[k]  += real_sample[n] * cos_coefficients_table[index];
            imag_op[k]  += real_sample[n] * sin_coefficients_table[index];
        }
    }
}
```

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
10503169	10503169	52.516 ms	52.516 ms	10503170	10503170	no

Detail

Instance

		Latency (cycles)		Latency (absolute)		Interval (cycles)		
Instance	Module	min	max	min	max	min	max	Type
grp_dft_Pipeline_VITIS_LOOP_19_2_fu_78	dft_Pipeline_VITIS_LOOP_19_2	10253	10253	51.265 us	51.265 us	10253	10253	no

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- VITIS_LOOP_17_1	10503168	10503168	10257	-	-	1024	no

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- VITIS_LOOP_19_2	10251	10251	22	10	1	1024	yes

In the baseline, Vitis HLS automatically do pipeline in the inner for loop, but can only achieve II = 10, because the inner for loop exist read after write data dependency, so the total latency approximately equals to 1024 x (22 + 10 x 1023 + 2) ≈ 1.05 x 10^7 cycles.

2. Opt1_pipeline

```
void dft(DTYPE real_sample[1024], DTYPE imag_sample[1024],DTYPE real_op[1024],DTYPE imag_op[1024])
{
    int k, n;
    int index;

    for (n = 0; n < 1024; n++)
    {
        for (k = 0; k < 1024; k++){
            #pragma HLS PIPELINE II=1
            index = (k * n) & 1023;
            real_op[k]  += real_sample[n] * cos_coefficients_table[index];
            imag_op[k]  += real_sample[n] * sin_coefficients_table[index];
        }
    }
    return;
}
```

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
1048601	1048601	6.081 ms	6.081 ms	1048602	1048602	no

Detail

Instance

N/A

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- VITIS_LOOP_10_1_VITIS_LOOP_12_2	1048599	1048599	25	1	1	1048576	yes

In opt1_pipeline, we change the order of the outer and inner for loop, therefore solve the data dependency problem and we can achieve II = 1 in the inner loop, and the total latency become ≈ 1/10x of the baseline.

3. Opt2_unroll

Factor = 16:

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
65570	65570	0.328 ms	0.328 ms	65571	65571	no

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- VITIS_LOOP_16_1_VITIS_LOOP_18_2	65568	65568	34	1	1	65536	yes

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	526	-
FIFO	-	-	-	-	-
Instance	-	161	18048	25660	-
Memory	620	-	0	0	-
Multiplexer	-	-	-	87	-
Register	-	-	18098	1248	-
Total	620	161	36146	27521	0
Available	280	220	106400	53200	0
Utilization (%)	221	73	33	51	0

Memory	Module	BRAM_18K	FF	LUT	URAM	Words	Bits	Banks	W*Bits*Banks
p_ZL22cos_coefficients_table_0_U	p_ZL22cos_coefficients_table_0_ROM_AUTO_1R	30	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_10_U	p_ZL22cos_coefficients_table_10_ROM_AUTO_1R	22	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_11_U	p_ZL22cos_coefficients_table_11_ROM_AUTO_1R	14	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_12_U	p_ZL22cos_coefficients_table_12_ROM_AUTO_1R	26	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_13_U	p_ZL22cos_coefficients_table_13_ROM_AUTO_1R	14	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_14_U	p_ZL22cos_coefficients_table_14_ROM_AUTO_1R	22	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_15_U	p_ZL22cos_coefficients_table_15_ROM_AUTO_1R	14	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_1_U	p_ZL22cos_coefficients_table_1_ROM_AUTO_1R	14	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_2_U	p_ZL22cos_coefficients_table_2_ROM_AUTO_1R	22	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_3_U	p_ZL22cos_coefficients_table_3_ROM_AUTO_1R	14	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_4_U	p_ZL22cos_coefficients_table_4_ROM_AUTO_1R	26	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_5_U	p_ZL22cos_coefficients_table_5_ROM_AUTO_1R	14	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_6_U	p_ZL22cos_coefficients_table_6_ROM_AUTO_1R	22	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_7_U	p_ZL22cos_coefficients_table_7_ROM_AUTO_1R	14	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_8_U	p_ZL22cos_coefficients_table_8_ROM_AUTO_1R	28	0	0	0	64	32	1	2048
p_ZL22cos_coefficients_table_9_U	p_ZL22cos_coefficients_table_9_ROM_AUTO_1R	14	0	0	0	64	32	1	2048

When applying array partition and loop unroll, Vitis HLS also do pipeline automatically, so the total latency improve massively. When the unroll factor is 16, the total latency ≈ 1/16 of opt1_pipeline, and if enhance the unroll factor, the latency improvement enhance linearly.

Here we can see that the BRAM usage exceeds the limit, because though we apply array partition on cos and sin table, but the access to cos and sin table are dependent on the loop index (n * k) & 1023, so the access are not sequentially, therefore need to store more duplicated coefficient in BRAM.

Copy cos and sin table manually:

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	280	-
FIFO	-	-	-	-	-
Instance	-	161	18048	23840	-
Memory	64	-	0	0	-
Multiplexer	-	-	-	87	-
Register	-	-	7114	1184	-
Total	64	161	25162	25391	0
Available	280	220	106400	53200	0
Utilization (%)	22	73	23	47	0

Memory	Module	BRAM_18K	FF	LUT	URAM	Words	Bits	Banks	W*Bits*Banks
cos_coefficients_table_0_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_1_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_2_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_3_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_4_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_5_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_6_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_7_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_8_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_9_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_10_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_11_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_12_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_13_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_14_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768
cos_coefficients_table_15_U	cos_coefficients_table_0_ROM_AUTO_1R	2	0	0	0	1024	32	1	32768

After copy the coefficient table manually, like we copy 16 cos and sin table here, then we can access 16 coefficients by 16 different array, and the total BRAM usage are only 64, for one cos or sin table only use 2 BRAM each, also the flip flop and LUT usage are less.

Factor = 32:

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	487	-
FIFO	-	-	-	-	-
Instance	-	321	36096	47680	-
Memory	128	-	0	0	-
Multiplexer	-	-	-	87	-
Register	-	-	13623	2208	-
Total	128	321	49719	50462	0
Available	280	220	106400	53200	0
Utilization (%)	45	145	46	94	0

fadd_32ns_32ns_32_10_full_dsp_1_U32	fadd_32ns_32ns_32_10_full_dsp_1	0	2	365	421	0
fmul_32ns_32ns_32_8_max_dsp_1_U33	fmul_32ns_32ns_32_8_max_dsp_1	0	3	199	324	0

When the partition and unroll factor enhance to 32, the hardware usage is the twice of the case factor 16, but the DSP usage exceeds the limit, since a floating point adder needs 2 DSPs and a floating point multiplier needs 3 DSPs, so the maximum factor will be 21.

Factor = 21:

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
50210	50210	0.281 ms	0.281 ms	50211	50211	no

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- VITIS_LOOP_18_1_VITIS_LOOP_20_2	50208	50208	34	1	1	50176	yes

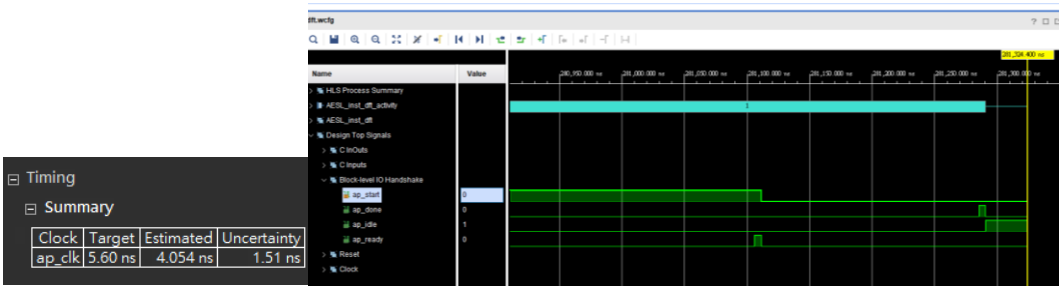
Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	1	-	-	-
Expression	-	-	0	677	-
FIFO	-	-	-	-	-
Instance	-	211	23604	30450	-
Memory	84	-	0	0	-
Multiplexer	-	-	-	72	-
Register	-	-	9125	1536	-
Total	84	212	32729	32735	0
Available	280	220	106400	53200	0
Utilization (%)	30	96	30	61	0

Finally, we assign array partition and unroll factor to be 21 and do the loop unroll by hand, and we get the best performance with latency $\approx 5\text{x}10^4$ cycles and also maximize the hardware utilization.

C. Comparison and result

	baseline	Opt1_pipeline	Opt2_unroll
Latency	10503169	1048601	50210
improvement	1x	10x	209x



When using array partition and loop unroll with factor = 21, we get the best performance, the clock cycle time is 5.6ns, and the maximum frequency is 246.67Mhz, and the total simulation time in co-sim is 281324.4ns, so our final score is 0.0049.

D. Github link: https://github.com/b07901049/1111HLS_labC_DFT.git