

RISC-V CON

ONLINE WEBINAR

TWS Solution Using Andes D25+ACE

Wei Tung
2020/04/23

Highlights of TWS with ACE

Function Name	Insn cycles	ACE cycles	Speedup
FIR (128 taps)	1692	128	12x
LMS (128 taps)	1820	128	14x
Data transfer	bus traffic	background	
ANC Hybrid Solution - Noise Control	10404	250	40x

Agenda

- ❖ TWS Platform
- ❖ Andes V5 Role in TWS
- ❖ D25 + ACE TWS Solution
- ❖ TWS ACE Instruction

TWS Platform Introduction

TrueWireless Stereo (TWS) Platform

Playback sync: TrueWireless™ Stereo

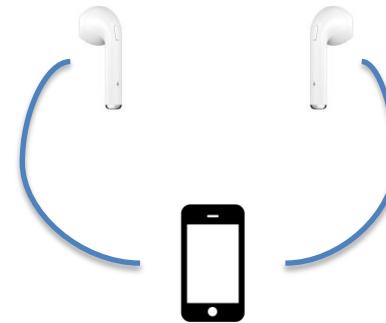
Bluetooth Version: Bluetooth 5.0, 4.2

Bluetooth Technology: Dual-mode, Single, BLE

ANC: Feedforward, Feedback, Hybrid

Audio interface: I2S, SPDIF, DAC

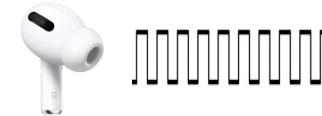
MIC: N * AMIC, N * DMIC



TWS Challenges

♦ Audio playback synchronization

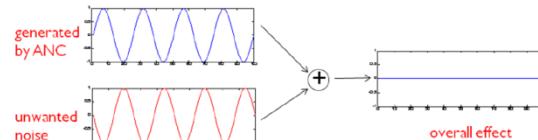
- PLL accuracy
- Playback start timing



♦ Active noise control computation ACE



- High amount of computation
- Low latency data path requirement



♦ Audio codec and audio format conversion

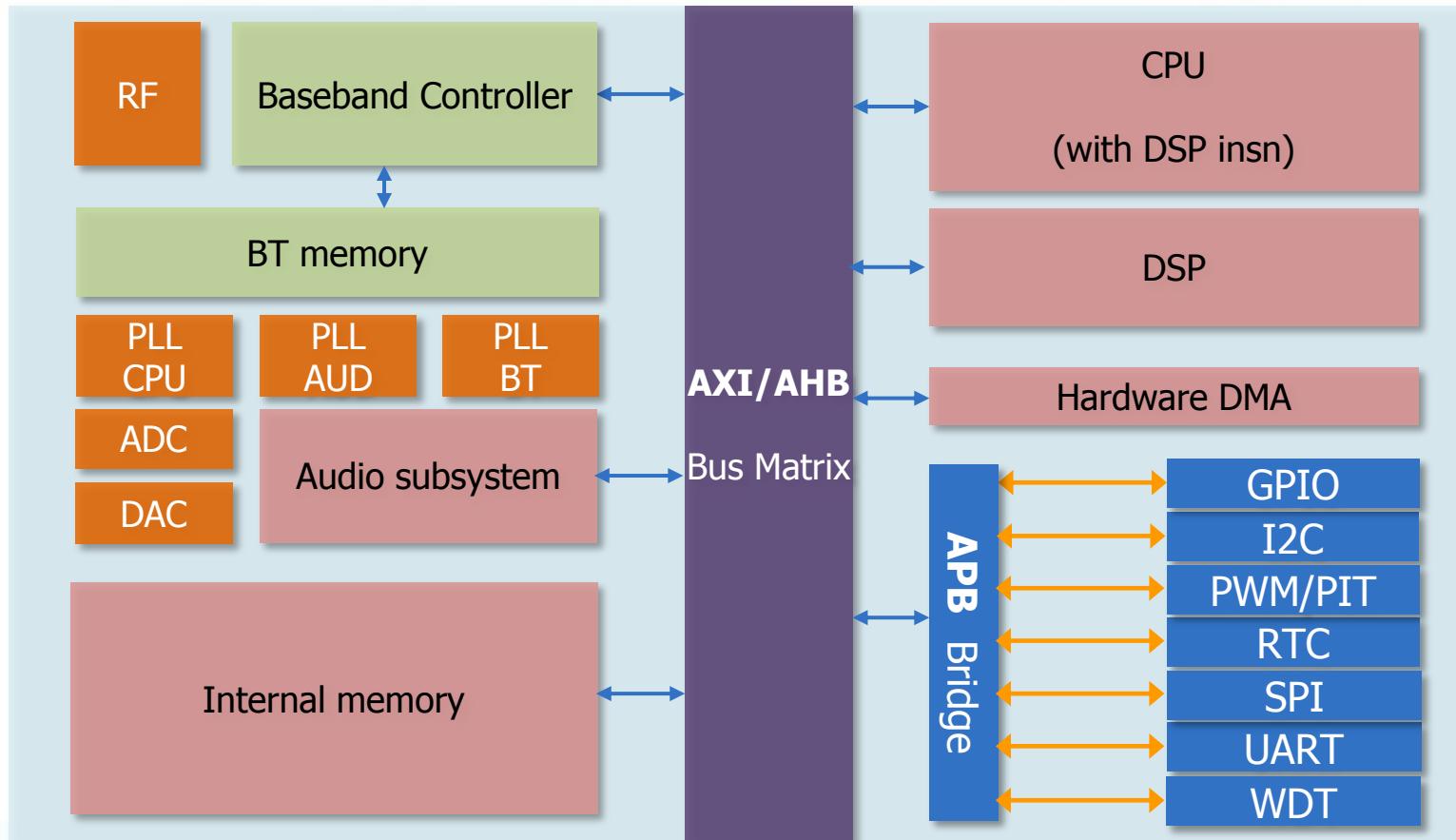
- PCM(24bits), little endian, big endian ACE
- LDAC, APTX, AAC



♦ Computation and power trade off

- Large amount of multiplication and addition ACE
- *Low power & high precision DAC (Analog)*
- *Low latency ADC*

TWS Platform



ANC Challenges in TWS

- ◆ ANC challenges
 - Restrict latency
 - Real time sample base streaming (1 sample in, 1 sample out)
 - Limited resource with high computation requirement
 - Low power, Low latency, High sample rate ADC & DAC
- ◆ Example:
 - ADC & DAC in 192KHz (1 sample = $\frac{1}{192KHz} \sim= 5\mu s$)
 - CPU clock 160MHz
 - The algorithm should be finished in 800 clock cycles. ($160M * 5\mu s = 800$)

ANC Algorithm (Hybrid)

- For higher noise control performance, the higher filter tap number should be used
- For example: how many addition and multiplication are needed for 128 taps ?
- Total **896** additions & **896** multiplications for **1** sample

ANC Conclusion

- ◆ Example:

- ADC & DAC in 192KHz (1 sample = $\frac{1}{192KHz} \sim= 5\mu s$)
- CPU clock 160MHz
- The algorithm should be finished in 800 clock cycles. ($160M * 5\mu s = 800$)
- ANC Feedforward : Total **384** MACs for **1** sample
- ANC Hybrid : Total **896** MACs for **1** sample

ANC FIR & LMS Profiling

Total Self Cycle Count: 22491						
Name	Calls	Self.InsC	Self.CvcC	Total InsC	Total CvcC	Time Percent...
fir_cal	4	5,148	6,765	5,148	6,765	30.08%
init_but	7	5,411	5,522	5,411	5,522	24.55%
do_printf	4	1,808	3,938	3,536	6,002	17.51%
lms_cal	2	2,826	3,639	2,826	3,639	16.18%
vprintf_help	68	1,156	1,426	1,564	1,846	6.34%
_riscv_save_3	68	408	420	408	420	1.87%
vprintf	4	124	159	3,688	6,215	0.71%
main	1	96	155	17,229	22,380	0.69%
strlen	4	88	124	88	124	0.55%
_riscv_save_12	4	76	94	76	94	0.42%
printf	4	60	84	3,748	6,299	0.37%
_write	4	28	54	28	54	0.24%
memset	1	25	43	38	60	0.19%
_start	1	24	36	17,299	22,491	0.16%
memset + 70	1	13	17	13	17	0.08%
_exit	1	8	15	8	15	0.07%

Single 128-tap FIR takes about **1600** cycles

Single 128-tap LMS takes about **1800** cycles

ANC in Hardware

Advantage

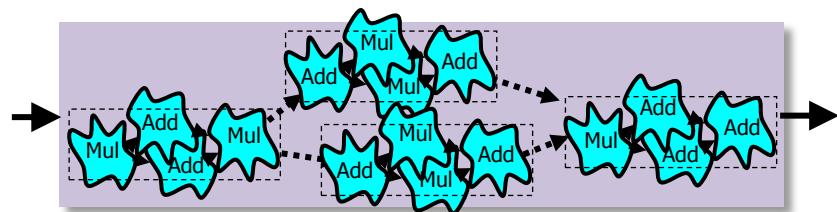
1. Parallel computing
2. Optimized data path

Absolute Efficiency **(not flexible)**

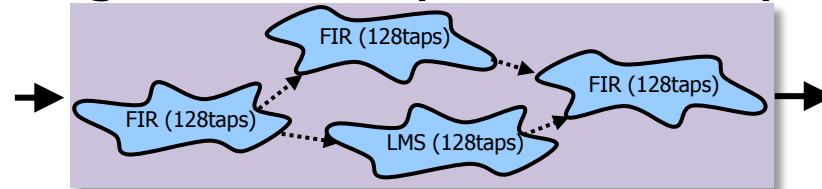


Disadvantage

1. Fixed architecture, Fixed algorithm
 2. Complicated SRAM access
- Complete Flexibility **(power consuming)**



High Efficiency & Flexibility

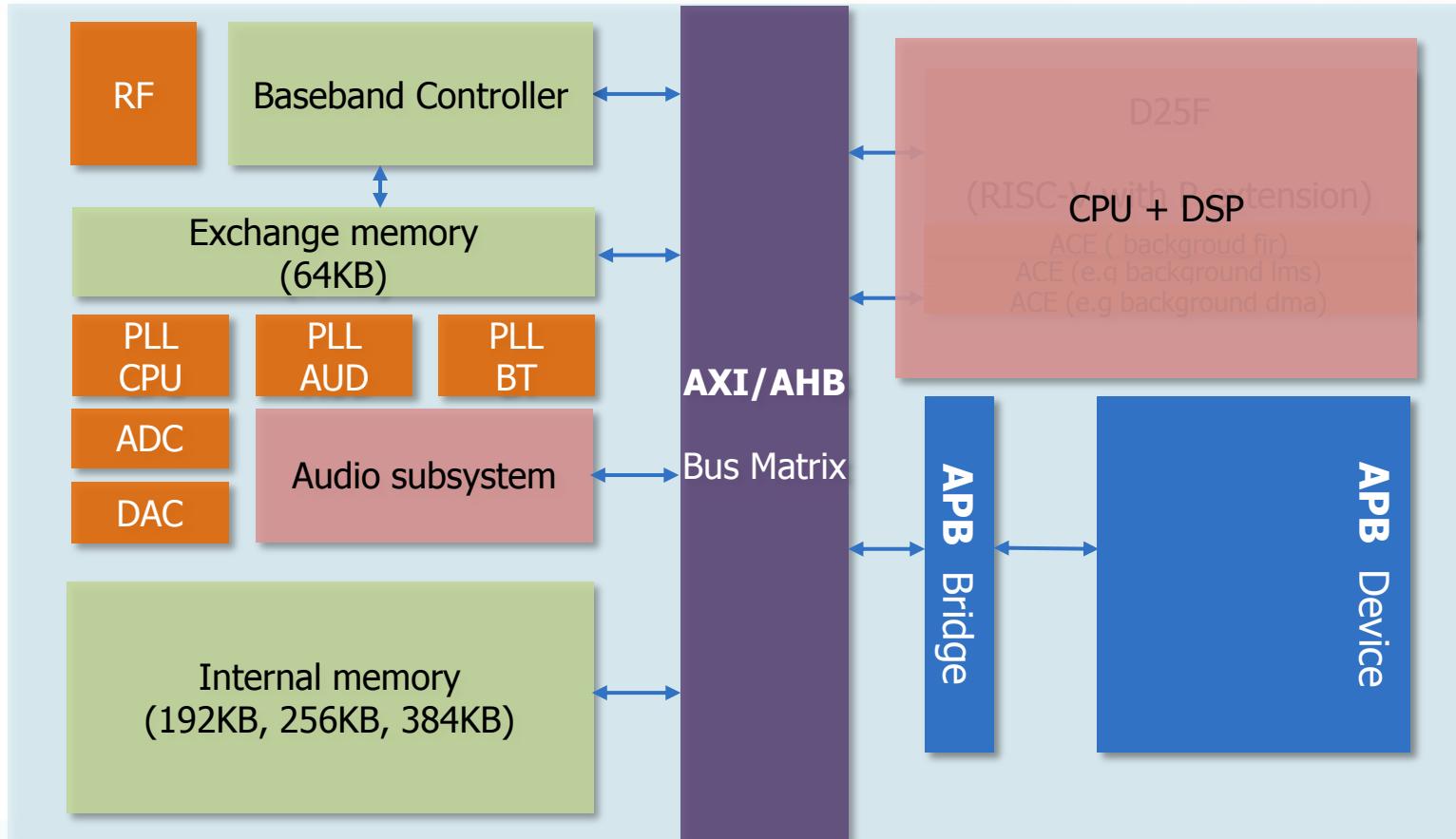


Andes V5 Role in TWS

Andes V5 Role in TWS

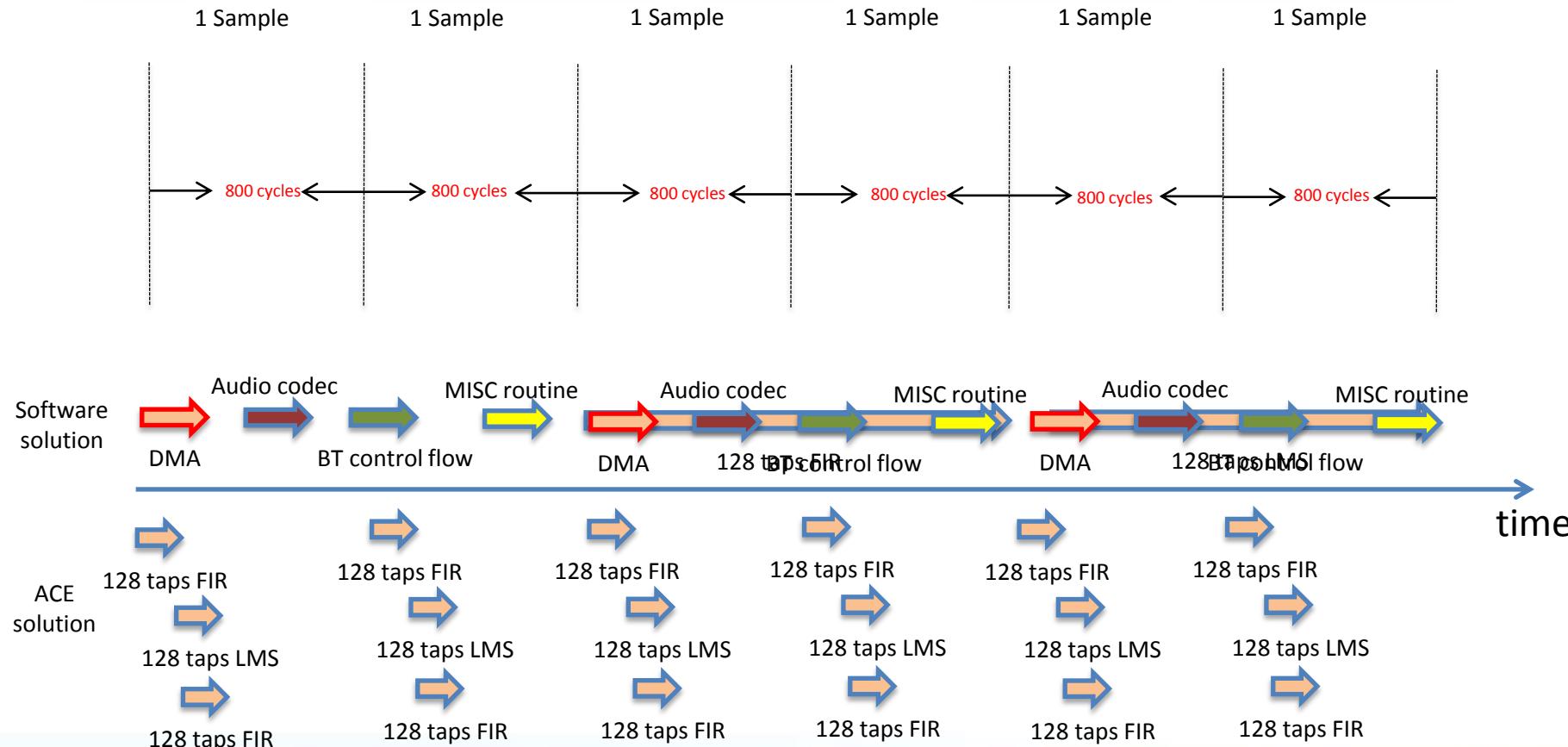
- CPU + DSP
 - Lots of redundant functions
- DSP + DSP
 - Lots of redundant functions
- D25 + ACE
 - **Best choice for TWS!**

TWS Platform



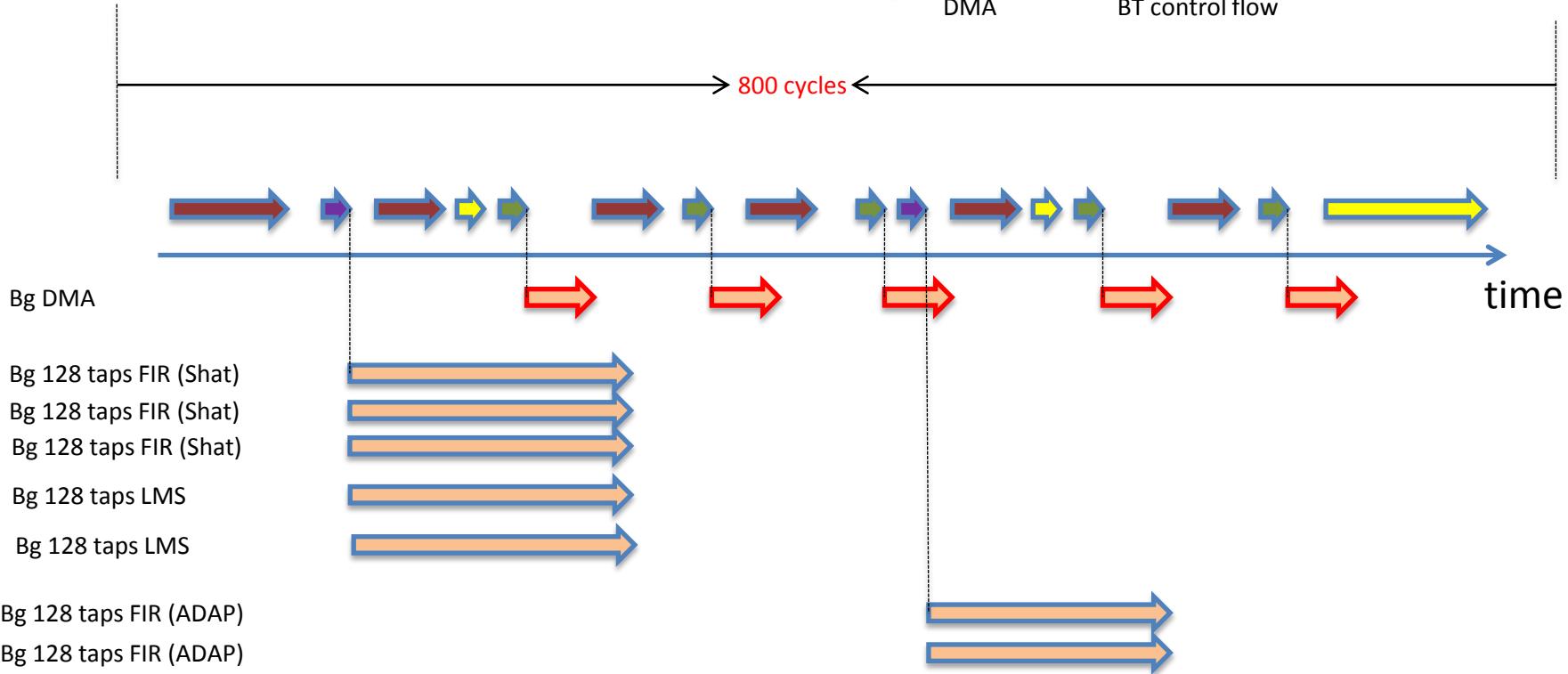
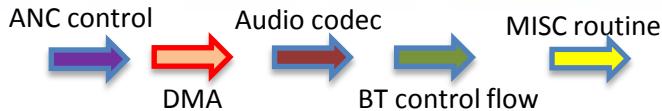
D25 + ACE TWS Solution

ANC: D25 W/ ACE Solution



ANC: D25 W/ ACE Solution

ANC Hybrid solution (low power, small area)



ANC: D25 W/ ACE Solution

- ◆ Example:

- ADC & DAC in 192KHz (1 sample = $\frac{1}{192KHz} \approx 5\mu s$)
- CPU clock 160MHz
- The algorithm should be finished in 800 clock cycles. ($160M * 5\mu s = 800$)

- ◆ Example (with ACE):

- ADC & DAC in 192KHz (1 sample = $\frac{1}{192KHz} \approx 5\mu s$)
- CPU clock **80MHz**
- The algorithm should be finished in **400** clock cycles. ($80M * 5\mu s = 400$)

TWS D25+ACE Solution

```
ram coef_sram {      //coef. Custom Memory
}
ram data_sram{       //vector Custom Memory
}
vec bg_insn fir_cal {
    operand=...;
    csim=...;
    latency= 128;      //enable multi-cycle ctrl
};
vec bg_insn lms_cal {
    operand=...;
    csim=...;
    latency= 128;      //enable multi-cycle ctrl
};
bg_insn bt_dma{
    operand=...;
    csim=...;
};
```

tws.ace

```
//ACE_BEGIN: fir_cal
reg          [31:0]  fir_out_reg;
wire         [31:0]  acc;

assign      acc = (ace_vstart) ? 64'd0 : fir_out_reg;
assign      fir_out = coef32[31:0] * buf32[31:0] + acc;

always @ (posedge core_clk or negedge core_reset_n)
begin
    ....
end
//ACE_END

//ACE_BEGIN: lms_cal
assign      coef32_out = ....;

//ACE_END

//ACE_BEGIN: bt_dma
....
//ACE_END
```

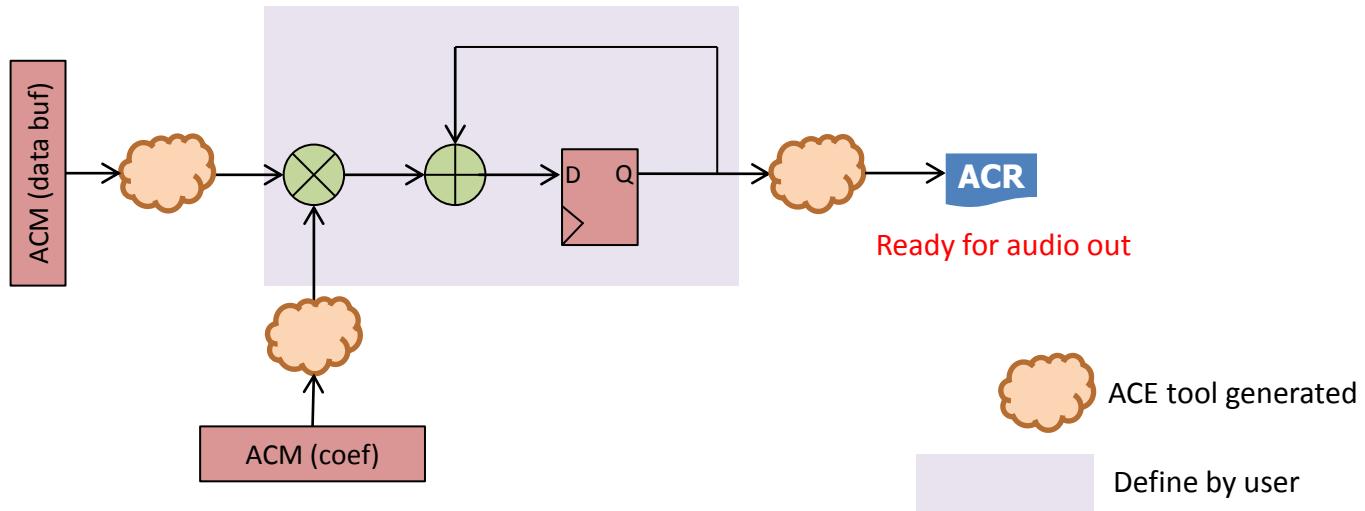
tws.v

TWS ACE Instruction Introduction

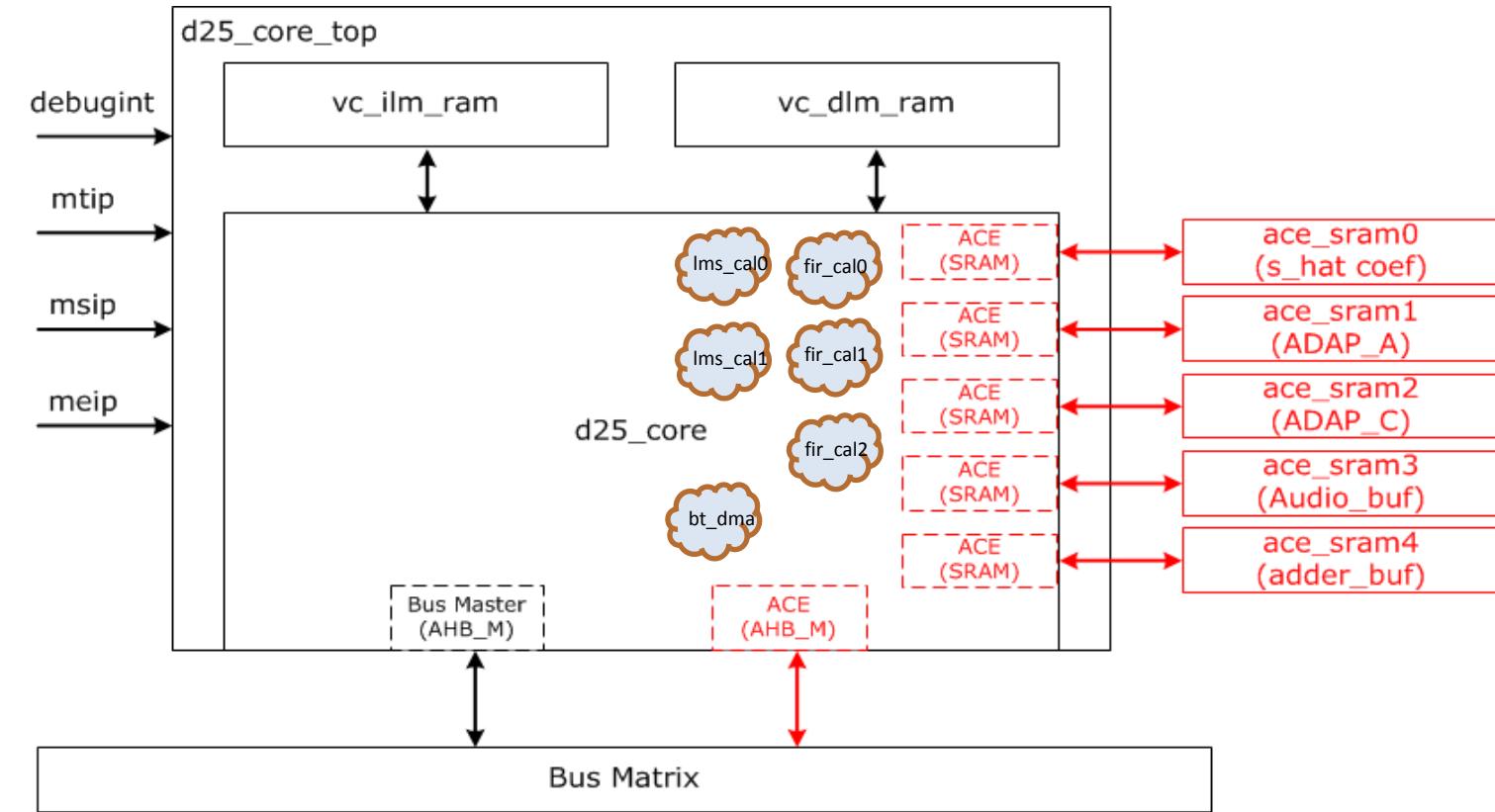
128-Tap FIR Instruction

$$W(z) \Rightarrow y[n] = \sum_{k=0}^M W_k * x[n - k]$$

```
for (int k = 0; k < M; k++)
    Y += (W[k] * X[k]); //calculate filter output
```



D25 ACE Platform



Highlights of TWS with ACE

Function Name	Insn cycles	ACE cycles	Speedup
FIR (128 taps)	1692	128	12x
LMS (128 taps)	1820	128	14x
Data transfer	bus traffic	background	
ANC Hybrid Solution - Noise Control	10404	250	40x

D25 + ACE @CPU clock **80MHz**

Conclusion

- TWS solution using Andes D25+ACE
- New idea of how to design a SOC
- Andes V5 role in the TWS platform
- Example of ANC customized instruction using ACE



RISC-V CON

ONLINE WEBINAR

Thank you,
See you next webinar!