HAU

硬件运算单元

目录

1 简介

HAU 硬件运算单元为弥补 M0 的运算处理能力而设计，包含硬件除法及硬件开方运算功能。

2 主要特征

- 运算时硬件自动暂停总线访问，不需要程序等待或查询运算进程
- 32 位除法器
- 支持无符号数/有符号数除法
- 支持 32/32 和 32/16
- 32/32 一次运算需要 34 个时钟周期；32/16 一次运算需 18 个时钟周期
- 32 位整数开方运算
- 开方运算需要 17 个周期

3 HAU 地址

HAU 与 CRC 共用 1KB 地址空间，连接在 AHB1 总线上，基地址为 0x4002_3000。

4 寄存器

DIV 模块共有 5 个 32 位寄存器，且必须按字访问。

DIV_CSR (DIV control and status register)

Address offset : 0x20

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME														DIV_D16		DIV_MODE
R/W	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	r	rw
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

NAME	DIV_DZE	DIV_OV				DIV_Z	DIV_C	DIV_N								
R/W	rw	rw	r	r	r	rw	rw	rw	r	r	r	r	r	r	r	r
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIV_MODE : DIV 模式

1 = 有符号

0 = 无符号

DIV_D16 : 32/16 计算

1 = 32/16

0 = 32/32

DIV_N ：余数符号位，由硬件置位。无符号运算时该位无效。

1 = 余数为负

0 = 余数为正

DIV_C ：中间变量，不应使用

DIV_Z ：余数 0 标志

1 = 余数为 0

0 = 余数为非 0

DIV_OV ：溢出标志，表示除法运算结果不可用。软件只能写 0。

1 = 运算结果溢出

0 = 无溢出

DIV_DZE ：除 0 错误标志，软件只能写 0。

1 = 除数为 0

0 = 除数为非 0

DIV_D (DIV dividend reg)

Address offset : 0x24

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	DIV_D[15:0]															
R/W	rw															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	DIV_D[31:16]															
R/W	rw															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIV_D[31:0]　　：被除数

DIV_S (DIV divisor reg)

Address offset : 0x28

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	DIV_S[15:0]															
R/W	rw															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	DIV_S[31:16]															
R/W	rw															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIV_S [31:0]　：除数

先写被除数，再写除数，除法运算会立即开始。

注：当 32/16 运算时，该寄存器只允许写入 16 位范围的整数。

DIV_R (DIV reminder reg)

Address offset : 0x2C

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	DIV_R[15:0]															
R/W	r															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	DIV_R[31:16]															
R/W	r															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIV_R [31:0]　：余数

DIV_Q (DIV quotient reg)

Address offset : 0x30

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NAME	DIV_Q[15:0]															
R/W	r															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	DIV_Q[31:16]															
R/W	r															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIV_S [31:0] ：商

开方运算包含 2 个寄存器。

SQRT_RADICAN (SQRT radican reg)

Address offset : 0x40

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	SQRT_RADICAN[15:0]															
R/W	rw															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	SQRT_RADICAN[31:16]															
R/W	rw															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SQRT_RADICAN [31:0] ：被开方数寄存器，对该寄存器写入后即开始进行开方运算。开方运算后该寄存器结果不可用。

SQRT_ROOT (SQRT ROOT reg)

Address offset : 0x44

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	SQRT_ROOT[15:0]															
R/W	r															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	-															
R/W	r															
reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SQRT_ROOT [15:0]：开方根寄存器。

5 HAU 编程

5.1 除法运算

1. 通过 DIV_CSR 配置除法（32/16 位、有/无符号运算）
2. 向 DIV_D 写入被除数
3. 向 DIV_S 写入除数
4. 从 DIV_Q 读取商；从 DIV_R 读取余数

5.2 开方运算

1. 向 SQRT_RADICAN 写入被开方数
2. 从 SQRT_ROOT 读取开方根