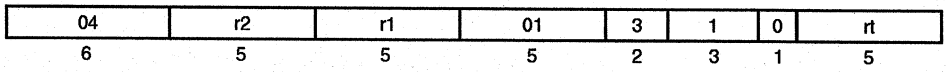
Appendix F Commodore PA-RISC Graphics SFU Instructions

GRAPHICS ARITHMETIC ON 5 BIT PIXELS **GR_ARITH_5**

Format:



04	r2	r1	01	3	1	0	rt
6	5	5	5	2	3	1	5

Purpose: To perform arithmetic on 16-bit High-Color pixels using integer arithmetic.

Description: GR pair r2 and GR pair r2 are used as the operands for the stated operation and the result is placed in GR pair rt. All register specifiers must point to an even register. Each 16-bit halfword of the 64 bits contained in the register pair is interpretted as a triplet of 5-bit fields, with the most significant bit of the halfword unused. When the arithemtic operation is performed, the carry between the bits separating the fields is broken thereby producing individual operations on each of the fields of four High-color pixels concurrently.

The operation performed on the fields is specified by the contents of Gr_R1. The least significant three bits specify the operation for the three fields of the triplet (i.e. bit 29 specifies the operation on the red field; bit 30 specifies the operation on the green field; bit 31 specifies the operation on the blue field). A zero indicates that an addition should be performed. A one inticates that a subtraction should be performed.

Each operation will saturate to the highest or lowest possible unsigned value as a function of the operation being performed.

Operation:

res[1:5]= sat(GR[r1[1:5]] op GR[r2[1:5]])

res[6:10]= sat(GR[r1[6:10]] op GR[r2[6:10]])

res[11:15]= sat(GR[r1[11:15]] op GR[r2[11:15]])

res[17:21]= sat(GR[r1[17:21]] op GR[r2[17:21]])

res[22:26]= sat(GR[r1[22:26]] op GR[r2[22:26]])

res[27:31]= sat(GR[r1[27:31]] op GR[r2[27:31]])

res[33:37]= sat(GR[r1[33:37]] op GR[r2[33:37]])

res[38:42]= sat(GR[r1[38:42]] op GR[r2[38:42]])

res[43:47]= sat(GR[r1[43:47]] op GR[r2[43:47]])

res[49:53]= sat(GR[r1[49:53]] op GR[r2[49:53]])

res[54:58]= sat(GR[r1[54:58]] op GR[r2[54:58]])

res[59:63]= sat(GR[r1[59:63]] op GR[r2[59:63]])

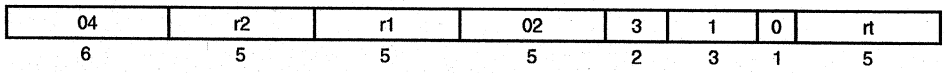
GR[t]= res

Exceptions: None

Notes:

GRAPHICS ARITHMETIC ON 5 BIT FIXED-POINT PIXELS — GR_ARITH_5_5

Format:



Purpose: To perform arithmetic on 16-bit High-Color pixels using fixed-point arithmetic.

Description: GR pair r2 and GR pair r1 are used as the operands for the stated operation and the result is placed in GR pair rt. All register specifiers must point to an even register. Each 32-bit fullword of the 64 bits contained in the register pair is interpretted as a triplet of 5.5-bit color fields, with the most significant two bits of the fullword unused. When the arithemtic operation is performed, the carry between the bits separating the fields is broken thereby producing individual operations on each of the fields of two High-color pixels concurrently.

The operation performed on the fields is specified by the contents of Gr_R1. The least significant three bits specify the operation for the three fields of the triplet (i.e. bit 29 specifies the operation on the red field; bit 30 specifies the operation on the green field; bit 31 specifies the operation on the blue field). A zero indicates that an addition should be performed. A one inticates that a subtraction should be performed.

Each operation will saturate to the highest or lowest possible unsigned value as a function of the operation being performed.

Operation:

res[2:11]= sat(GR[r1[2:11]] op GR[r2[2:11]])

res[12:21]= sat(GR[r1[12:21]] op GR[r2[12:21]])

res[22:31]= sat(GR[r1[22:31]] op GR[r2[22:31]])

res[34:43]= sat(GR[r1[34:43]] op GR[r2[34:43]])

res[44:53]= sat(GR[r1[44:53]] op GR[r2[44:53]])

res[54:63]= sat(GR[r1[54:63]] op GR[r2[54:63]])

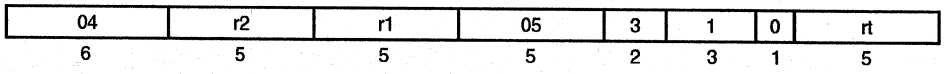
GR[t]= res

Exceptions: None

Notes:

GRAPHICS ARITHEMETIC ON 8 BIT FIXED-POINT PIXELS — GR_ARITH_8_8

Format:



Purpose: To perform arithmetic on 8-bit True-Color pixels using fixed-point arithmetic.

Description: GR pair r2 and GR pair r1 are used as the operands for the stated operation and the result is placed in GR pair rt. All register specifiers must point to an even register. Each 16-bit halfword of the 64 bits contained in the register pair is interpretted as an 8.8 true color pixel. When the arithemtic operation is performed, the carry between the bits separating the fields is broken thereby producing individual operations on each of the four True-color pixels concurrently.

The operation performed on the fields is specified by the contents of Gr_R1. The least significant bit specifies the operation to be performed on the pixels. A zero indicates that an addition should be performed. A one inticates that a subtraction should be performed.

Each operation will saturate to the highest or lowest possible unsigned value as a function of the operation being performed.

Operation:

res[0:15]= sat(GR[r1[0:15]] op GR[r2[0:15]])

res[16:31]= sat(GR[r1[16:31]] op GR[r2[16:31]])

res[32:47]= sat(GR[r1[32:47]] op GR[r2[32:47]])

res[48:63]= sat(GR[r1[48:63]] op GR[r2[48:63]])

GR[t]= res

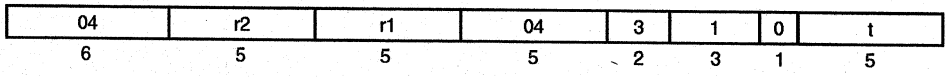
Exceptions: None

Notes:

GRAPHICS CLIP SUB

GR_CLIP_SUB

Format:



Purpose: To subtract two register pairs and save the outcodes in a condition shift register.

Description: GR pair r2 is subtracted from GR pair r1 and the result is placed in GR pair t. Both register specifiers must point to an even register. The 64 bits contained in the register pair are interpretted as a coordinate pair. The most significant 32 bits is the X coordinate; the least significant 32 bits is the Y coordinate. The carry between the X and Y portions is broken during the operation. The carries out of bits 0 and 32 are shifted into two 4-bit shift registers for later use.

This instruction is intended to be used for outcode pre-clipping.

Operation: res[0:311]= sat(GR[r1[0:31]] + GR[r2[0:31]])

res[32:63]= sat(GR[r1[32:63]] + GR[r2[32:63]])

GR[t]= res

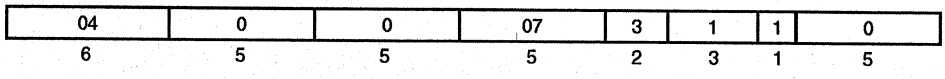
Exceptions: None

Notes:

GRAPHICS CLIP TEST

GR_CLIP_TEST

Format:



Purpose: To test the result of earlier graphics clip subtracts by evaluating the resulting outcodes.

Description: The two 4-bit shift registers used to collect the results of the GR_CLIP_SUB instruction are evaluated. If the resulting outcodes indicate that preclipping has occurred, then the following instruction is executed. Otherwise, the following instruction is skipped.

This instruction is intended to be used for outcode pre-clipping.

Operation:

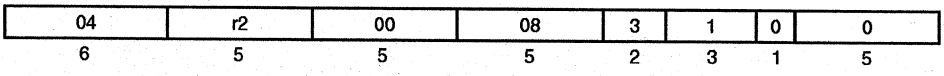
```
if(outcode_shift_reg)
    PSW[N] = 1;
```

Exceptions: None

Notes:

RAPHICS MOVE TO CONTROL REGISTER GR_MTCL

Format:



04	r2	00	08	3	1	0	0
6	5	5	5	2	3	1	5

Purpose: To move a value from a general register into the graphics control register Gr_R1.

Description: The value in GR[r2] is moved into the graphics control register Gr_R1. It is a three bit register which controls the operation performed during graphics arithmetic instructions.

Operation: Gr_R1= GR[r2[29:21]]

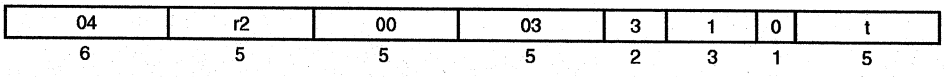
Exceptions: None

Notes:

GRAPHICS PACK 5 BIT FIXED-POINT PIXELS

GR_PACK_5_5

Format:



04	r2	00	03	3	1	0	t
6	5	5	5	2	3	1	5

Purpose: To pack two 16-bit fixed-point represented High-Color pixels into a register.

Description: Two High-color pixels are extracted from GR pair r2 and placed in GR pair t. Register specifier r2 must point to an even register. Each 32-bit fullword of the 64 bits contained in the register pair is interpretted as a triplet of 5.5-bit color fields, with the most significant two bits of the fullword unused. The pack moves the integer portion of each fixed point number into the appropriate position of the result register.

Operation:

res[1:5]= GR[r2[2:6]]

res[6:10]= GR[r2[12:16]]

res[11:15]= GR[r2[22:26]]

res[17:21]= GR[r2[34:38]]

res[22:26]= GR[r2[44:48]]

res[27:31]= GR[r2[54:58]]

GR[t]= res

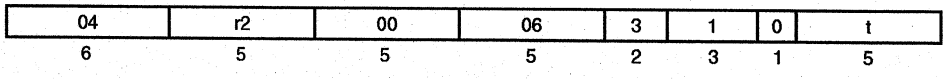
Exceptions: None

Notes:

GRAPHICS PACK 8 BIT FIXED-POINT PIXELS

GR_PACK_8_8

Format:



Purpose: To pack four 8-bit fixed-point represented True-Color pixels into a register.

Description: Four True-color pixels are extracted from GR pair r2 and placed in GR pair t. Register specifier r2 must point to an even register. Each 16-bit halfword of the 64 bits contained in the register pair is interpretted as an 8.8 true-color field. The pack moves the integer portion of each fixed point number into the appropriate position of the result register.

Operation:

res[0:7]= GR[r2[0:7]]

res[8:15]= GR[r2[16:23]]

res[16:23]= GR[r2[32:39]]

res[24:31]= GR[r2[48:55]]

GR[t]= res

Exceptions: None

Notes: