- rra.s o' arra.s
- tomic counters
- $ma&es
- Separate pro&ram ob6ects @also 7no#n as separate shader ob6ectsA
-

$

compute shader has access to man. o' the same resources as 'ra&ment and other shader processors, such as te2tures, bu''ers, ima&e variables, atomic counters, and so on. $t does not have an. prede'ined inputs nor an. 'i2ed;'unction outputs. $t is not parE PA ri ê ðc 1 € c:E ¡€'pQ ¢ £ ¤ ¥ ¦ no E0• ` thes, A @esS d § ¨ s o

,

-he source character set used 'or the OpenGL ES shadin& lan&ua&es is 9nicode in the 9-4;5 encodin& scheme. $nvalid 9-4;5 characters are i&nored. ! urin& pre;processin&, the 'ollo#in& applies:

- b.te #ith the value Eero is al#a.s interpreted as the end o' the strin&
- "ac7slash @FJA, is used to indicate line continuation #hen immediatel. precedin& a ne#;line.
- >hite space consists o' one or more o' the 'ollo#in& characters: the space character, horiEontal cpEu,PÀ%@E(!@e,3H

! !

! !

) !

3# % % 3O9 : ! *3#* %

!!"#$%!!
!!&#"%!!

*			. &
1 @hi&hestA	parenthetical &roupin&	@ A	:
)	unar.	de'ined N ; O P	Ri&ht to Le't
(	multiplicative	Q ? R	Le't to Ri&ht
*	additive	N ;	Le't to Ri&ht
/	bit;#ise shi't	SS TT	Le't to Ri&ht
0	relational	S T SU TU	Le't to Ri&ht
8	e=ualit.	UU PU	Le't to Ri&ht
5	bit;#ise and	V	Le't to Ri&ht
6	bit;#ise e2clusive or	W	Le't to Ri&ht
1+	bit;#ise inclusive or	X	Le't to Ri&ht
11	lo&ical and	VV	Le't to Ri&ht
1) @lo#estA	lo&ical inclusive or	X X	Le't to Ri&ht

-he operator can be used in either o' the 'ollo#in& #a.s:

--here are no number si&n based operators @e.&. no or =A, no MM operator, nor is there a operator.

-he semantics o' appl.in& operators in the preprocessor match those standard in the %NN preprocessor #ith the 'ollo#in& e2ceptions:

- -he)nd operand in a lo&ical and @FVVFA operation is evaluated i' and onl. i' the 1st operand evaluates to non;Eero.
- -he)nd operand in a lo&ical or @FXXFA operation is evaluated i' and onl. i' the 1st operand evaluates to Eero.
- -here is no boolean t.pe and no boolean literals. or ! result is returned as inte&er or 5 respectivel.. > herever a boolean operand is e2pected, an. non;Eero inte&er is interpreted as and a Eero inte&er as !.

$' an operand is not evaluated, the presence o' unde'ined identi'iers in the operand #ill not cause an error.

3reprocessor e2pressions #ill be evaluated at compile time.

#ill cause the implementation to put a dia&nostic messa&e into the shader ob@ectKs in'ormation lo& @see section 8.1) HShader, 3ro&ram and 3ro&ram 3ipeline QueriesI in the OpenGL ES Graphics S.stem Speci'ication 'or ho# to access a shader ob@ectKs in'ormation lo&A. -he messa&e #ill be the to7ens 'ollo#in& the

! !

-he initial state o' the compiler is as i' the directive

/

#as issued, tellin& the compiler that all error and #arnin& reportin& must be done accordin& to this speci'ication, i&norin& an. e2tensions.

Each e2tension can de'ine its allo#ed &ranularit. o' scope. $' nothin& is said, the &ranularit. is a shader @that is, a sin&le compilation unitA, and the e2tension directives must occur be'ore an. non;preprocessor

! !

0

! !

H H H

<? <?> B <?! <?! > B

<? <?! <? <?!

E? E? > B E? E?

H H H

E? >! E? >! E? >!

) : one o'
$ < E ; F X Y Z [%

$denti'iers startin& #ith H&lBI are reserved 'or use b. OpenGL ES, and ma. not be declared in a shader. $t is an error to redeclare a variable, includin& those startin& H&lBI.

-he ma2imum len&th o' an identi'ier is 1+)* characters. $t is an error i' the len&th e2ceeds this value.

5 -

Some lan&ua&e rules described belo# depend on the 'ollo#in& de'initions.

5ò 5 › 0

shader contains a o' a variable . i', a'ter preprocessin&, the shader contains a statement that #ould read or #rite . @or part o' 2A, #hether or not run;time 'lo# o' contro Rnrtir

! !

! (&

ll variables and 'unctions must be declared be'ore bein& used. 1ariable and 'unction names are

& 9

E?

$ %& ! '(!

decimal;constant:
nonEero;di&it
decimal;constant di&it

! $! !)

.) !
$ # .) !)
$V # .) !)
.) ! # .) !)

)
$ 5)

5) one o'
< E ; F X Y Z [%

!) 8 one o'

E2amples

> *?2 –) 4 >2

> *?2 –@ 4 >2

5> *?2 –@ ; 2
(– 4 5>

$ %& ! '(!

! ' (

-he OpenGL ES Shadin& Lan&ua&e includes data t.pes 'or &eneric);, (;, and *;component vectors o' 'loatin&;point values, inte&ers, and "ooleans. 4loatin&;point vector variables can be used to store colors, normals, positions, te2ture coordinates, te2ture loo7up results and the li7e. "oolean vectors can be used 'or component;#ise comparisons o' numeric vectors. Some e2amples o' vector declarations are:

```
:          >4          :
=                :
```

"ecause a sin&le opa=ue t.pe declaration e''ectivel. declares t#o ob0ects, the opa=ue handle itsel' and the ob0ect it is a handle to, there is room 'or both a stora&e =uali'ier and a memor. =uali'ier. -he stora&e =uali'ier #ill =uali'. the opa=ue handle, #hile the memor. =uali'ier #ill =uali'. the ob0ect it is a handle to.

! 2

Sampler t.pes @e.&., E?A are opa=ue t.pes, declared and behavin& as described above 'or opa=ue t.pes.

! 4

n. restrictions on the usa&e o' a t.pe or =uali'ier also appl. to a structure that contains that t.pe or =uali'ier. -his applies recursivel..

Structures can contain variables o' an. t.pe e2cept:

- atomicBuint @since there is no mechanism to speci'. the bindin&A
- ima&e t.pes @since there is no mechanism to speci'. the 'ormat =uali'ierA

! 5 . &

1ariables o' the same t.pe can be a&&re&ated into arra.s b. declarin& a name 'ollo#ed b. brac7ets @,-A enclosin& an optional siEe. > hen present, t–p`wdp s oURRm si h S `e&!P`Ed` oeepesb`&

I=J
9 K I9 J
" 1 :
I " J

9 I=JI:J
2 =
I J2 :

<4 92=2D 7 '
0
IJ

```
 IJ 1      I:J  >2A4 :2A                          ;      .  :
;IJ 1      IJ  >2A4 :2A4 =2A                      ;      .  =

 IGJ
 IJ 1
```

: ote that the initialiEer itsel' does not need to be a constant e2pression but the len&th o' the initialiEer #ill be a constant e2pression.

rra.s can have initialiEers 'ormed 'rom arra. constructors:

```
IGJ 1      IGJ =294 92:4 G2A4 G2:4 >2>
IGJ 1      IJ =294 92:4 G2A4 G2:4 >2>
```

n arra. o' arra.s can be declared as

) 0 6

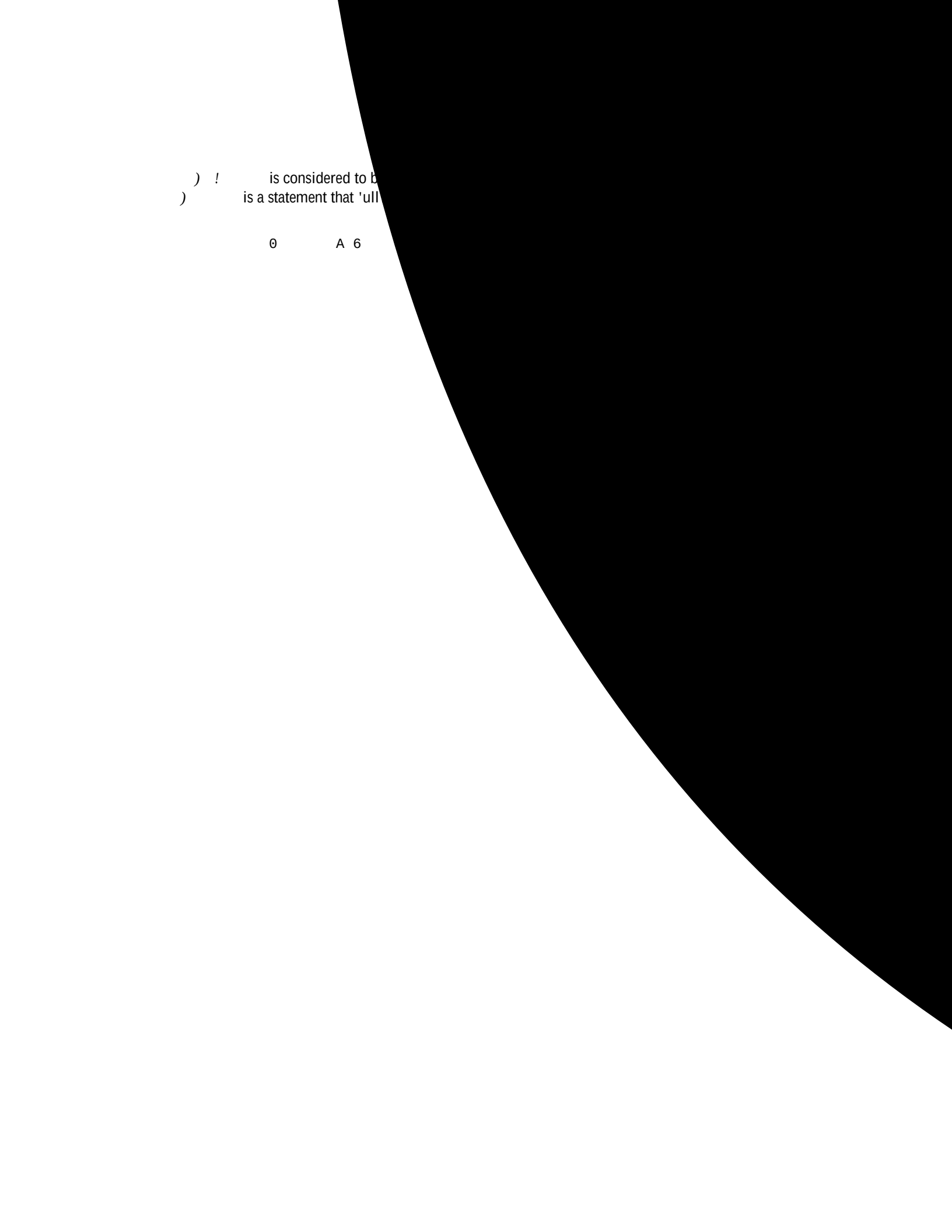
)) 7 <>
0
9 IJ
9 IJ 4 5 . ;
6

)) 7 <:
0
9 I9J 4 .
) IJ 4 5 . ; 4
;
6

% de'ines #here names ma. be de'ined. > ithin a sin&le % * a name has at most one

i';e2pression i';statement else;statement,

a variable declared in the i';statement is scoped to the end o' the i';statement. variable declared in the



$ %& ! '(!

! !

$ %& ! '(!

! ;

$ %& ! '(!

$ %& ! '(!

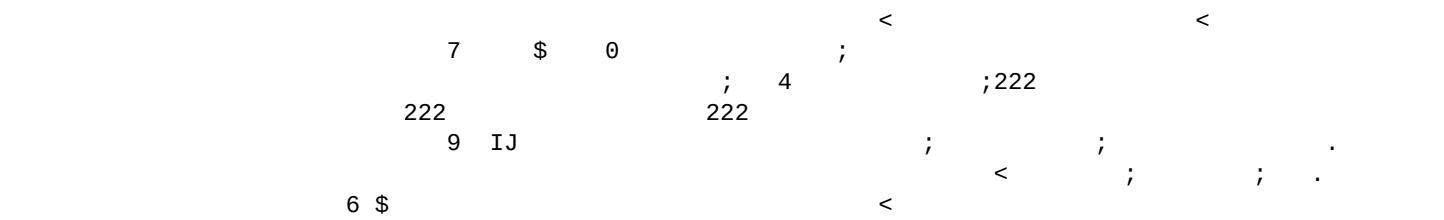
!)

.% is one o'

- / ! %
- n % %
- n ar

-he output o' the verte2 shader and the input o' the 'ra&ment shader 'orm a shader inter'ace. 4or this inter'ace, verte2 shader output variables and 'ra&ment shader input variables o' the same name must match in t.pe and =uali'ication, #ith a 'e# e2ceptions: -he stora&e =uali'iers must, o' course, di''er @one is and one is A. ##s op

$ %& ! '(!



-here are implementation;dependent limits on the number o' the shader stora&e bloc7s used 'or each t.pe o' shader, the combined number o' shader stora&e bloc7s used 'or a pro&ram, and the amount o' stora&e re=uired b. each individual shader stora&e bloc7. $' an. o' these limits are e2ceeded, it #ill cause a compile;time or lin7;time error.

GLSL ES (.1 does not support inter'ace bloc7s 'or shader inputs or outputs.

n inter'ace bloc7 is started b. a or 7e.#ord, 'ollo#ed b. a bloc7 name, 'ollo#ed b. an

Repeatin& the or inter'ace =uali'ier 'or a memberís stora&e =uali'ier is optional. 4or e2ample,

```
          ,
0
                  9     !          4                    <2
          9   8                    4                    <2
```

$' an instance name @ A is not used,

- a variable outside a bloc7, and a bloc7 #ith no instance name, #here the variable has the same name as a member in the bloc7.

4or bloc7s declared as arra.s, the arra. inde2 must also be included #hen accessin& members, as in this e2ample

```
          ,          0     BK#        ,         I:J                        :
         9            H    '   H
         9            H    '   K  8      H
                      -
  6           I9J
  222
```

$ %& ! '(!

$ %& ! '(!

; ;! !

Speci'.in& this #ill ma7e per;'ra&ment tests be per'ormed be'ore 'ra&ment shader e2ecution. $n addition it is an error to staticall. #rite to &lB4ra& ! epth in the 'ra&ment shader. $' this is not declared, per;

$ %& ! '(!

-he location speci'ies the location b. #hich the OpenGL ES 3$ can re'erence the uni'orm and update its

```
;          4       !  8
;          4       !  8
```

9ni'orm and shader stora&e bloc7s can be declared #ith optional la.out =uali'iers, and so can their individual member declarations. Such bloc7 la.out =uali'ication is scoped onl. to the content o' the

-he /) =uali'ier speci'ies the bindin& point correspondin& to the uni'orm or shader stora&e bloc7, #hich #ill be used to obtain the values o' the member variables o' the bloc7. $t is a compile;time error to speci'. the /)

```
;              1 :                !
```

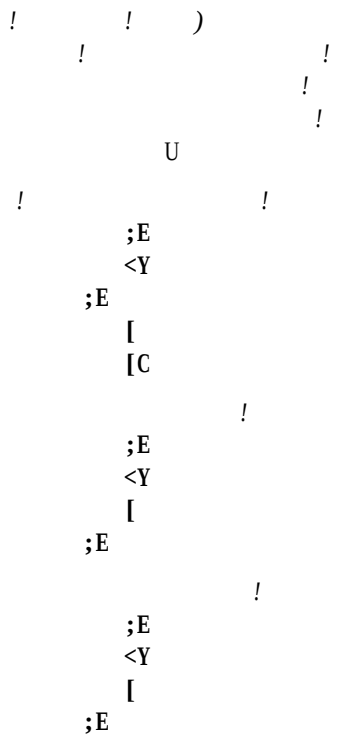
#ill establish that the atomic counter / has a bindin& to bu''er bindin& point) at an o''set o' 5 basic machine units into that bu''er. -he o''set 'or bindin& point) #ill a&ain be post;incremented b. * @the siEe o' an atomic counterA.

>hen multiple variables are listed in a la.out declaration, the e''ect #ill be the same as i' the. #ere declared one at a time, in order 'rom le't to ri&ht.

"indin& points are not inherited, onl. o''sets. Each bindin& point trac7s its o#n current de'ault 'or inheritance o' subse=uent variables usin& the same /) . -he initial state o' compilation is that all bindin& points have an o' +. -he can be set per bindin& point at &lobal scope @#ithout declarin& a variableA. 4or e2ample,

```
;              1 :4        1 9                !
```

Establishes that the ne2t C



$ %& ! '(!

3recisions are e2pressed in terms o' ma2imum relative error in units o' 9L3 @units in the last placeA,

-he re=uired ran&es and precisions 'or precision =uali'iers are:

; + + * + *
* : 9 * :
:
6

- 4or si&ned and unsi&ned inte&ers, the value is truncatedL bits in positions not present in the tar&et precision are set to Eero. @3ositions start at Eero and the least si&ni'icant bit is considered to be position Eero 'or this purpose.A

$n other

=-
N)
:-)
:-B ;
:-B ;)
:-H)

:-
0Ð

$nitiall., b. de'ault, all output variables are allo#ed to be variant. -o 'orce all output variables to be invariant, use the pra&ma

),-+"

be'ore all declarations in a shader. $' this pra&ma is used a'ter the declaration o' an. variables or 'unctions, then the set o' outputs that behave as invariant is unde'ined.

Generall., invariance is ensured at the cost o' 'le2ibilit. in optimiEation, so per'ormance can be de&raded

! 4)

$nvariance must be &uaranteed 'or constant e2pressions. particular constant e2pression must evaluate to

$ %& ! '(!

$ma&e variables =uali'ied #ith _ _ _ , or **B** ma. not be passed to

$ %& ! '(!!

')

!) !! !

' . &

-he constructor . / preserves the bit pattern in the ar&ument, #hich #ill chan&e the ar&ument's value i' its si&n bit is set. -he constructor . / preserves the bit pattern in the ar&ument, #hich #ill chan&e its value i' it is ne&ative.

Some use'ul vector constructors are as 'ollo#s:

```
=                    .                          =
9     9        <        9              5
9    :              9          A          ;        >

:      4                                 .        :      :
 =    4    4                             .          =      =
 9    4    4      4                  9 7

:    =                                                 =
=    9                                                  9

=    :4              =2  1    :2 4    =2; 1    :2;4    =2. 1
=      4    :         =2  1      4    =2; 1    :2 4    =2. 1    :2;
9    =4
9      4    =
9    :4    :
```

Some e2amples o' these are:

```
9       1    9 A2A4 >2A4 A2A4 >2A
9       1    9 >2A                          >2A
=       1    =                    9
```

-o initialiEe the dia&onal o' a matri2 #ith all other elements set to Eero:

```
:
=
9
```

-hat is, *! 7 87‹8* is set to the 'loat ar&ument 'or all *> ‹*

!) !! !

!) !! !

'!! . &

!) !! !

:
2
2.

rra. subscriptin& s.nta2 can also be applied to vectors to provide numeric inde2in&. So in

9

% 7 8 re'ers to the third element o' pos and is e=uivalent to % 5

!) !! !

#here the &eneral e2pression

5 1

is e=uivalent to

1

#here *! !* is the value returned b. *! !* .% * % is as described belo#, and the *! !* .% and .% must satis'. the semantic re=uirements o' both % and e=uals @**6**A.

- -he operator is multipl. @(A, #here both operands are matrices or one operand is a vector

!) !! !

.

!) !! !

' " (9)

0

%!
) !
.%

* !

* !

0 +

* !

___

_____

_____

_____ !

% _____ !

_____

0

%onditional control 'Io# in the shadin& lan&ua&e is done b. either , ; , or **B** statements:

```
!
    @/ ! .%      A
    @/ ! .%      A
    B  @    .%     AZ 3 #        ! % [
```

> here

0

4or, #hile, and do loops are allo#ed as 'ollo#s:

5 5 5
5

5
5

5

See section 6 HShadin& Lan&ua&e GrammarI 'or the de'initive speci'ication o' loops.

-he loop 'irst evaluates the .% , then the) .% . $' the) .% evaluates to

* !

0 ! =

2

2

4ra&ment shader helper invocations e2ecute the same shader code as non;helper invocations, but #ill not

- he built;in constant *!0 J ,& %(5*

!H ' * ' 1 >E

&)	-
&en4-.pe @&en4-.pe .A	Returns the h.perbolic sine 'unction

4 ! + 8* * 1 6 1 +

& + , !

&)

& + , !

&)

&) -

&en4-.pe

4 0 9) +

&) - g)

4 2 (: +

Relational and e=ualit. op=u10t. op=u10tb !

4 4 +

&) -

&en$-.pe K @&en$-.pe ! , E2tracts bits thrðu&h
int , int / AL

&en9-.pe K @&en9-.pe ! ,
int , int / AL

&) -

lo#p &en$-.pe **A>H**@&en$-.pe *!* AL Returns the bit number o' the least si&ni'icant one bit in
lo#p &en$-.pe **A>H**@&en9-.pe *!* AL

4 5) +

&) -

hi&hp ivec) > @&sampler) ! %! , int !)

& + , !

&)

&) -

```
&vec*        S  I@    @&sampler) !     %!  * vec( -,
                       vec) ) - ) ., vec) ) - ) A

&vec*        S  I@    @&sampler) !     %!  * vec* -,
                       vec) ) - ) ., vec) ) - ) A

&vec*        S  I@    @&sampler( !     %!  * vec* -,
                       vec( ) - ) ., vec( ) - ) A

'loat        S  I@    @samp,pĉ)
```

& + , !

&) - !

4 ! +

-- he shader invocation control 'unction is onl. available in compute shaders. $t is used to control the relative e2ecution order o' multiple shader invocations used to process a local #or7 &roup @in the case o' compnvotc0Pshadr i

dditionall., memor. barrier 'unctions order stores per'ormed b. the callin& invocation, as observed b.

5 9

d ',gm k s described in chapter 8 o' the OpenGL ES speci'ication, shadersâ ma. be lin7eed

P•À@•!pÂóÂ``6V

" -

- he precision o' a verte2 output does not need to match the precision o' the correspondin& 'ra&ment input.
- he minimum precision at #hich verte2 outputs are interpolated is the minimum o' the verte2 output

"

-

5 *

Quali'ier %lass	Quali'ier	?	9ni'orms	9"O
Stora&e		:?	Des	Des
u2iliar.		:o	:?	:?
		Des	Des	

La.out

GLSL S.nta2

~~ ternar. or lo&ical operation is onl. &uaranteed to be conver&ent a'ter the statement, irrespective o' the uni'ormit. o' the result o' the operation. -he e''ect o' this is that an e2pression such as~~

~~BXX7~~

~~0~~

~~0~~

' & . & & ' '

#

* *

"

-

"%&,!KB(%$ (#+Q,!KB(%$ "%&,!7(BN?%, (#+Q,!7(BN?%, "%&,!7(BN% (#+Q,!7(BN% -*,
N*HHB N*"*$ %[@B")%H#N*"*$ 7B$+ -B)Q ,#"-% K"@)),B( )"B)Q K%(N%$,
"%&,!B$+"% (#+Q,!B$+"% '%(,#NB"!7B(NB(%, BHK%()B$- [@%),#*$

-he 'olIo#in& describes the &rammar 'or the OpenGL ES Shadin& Lan&ua&e in terms o' the above to7ens.

```
    /! 0 )
    1  '2"141'

%      0 .%
       /! 0 )
    12"+$2("A2"
    N12"+$2("A2"
    4  $A"+$2("A2"
    B$$  +$2("A2"
     '4"0-A  '2  .%        1&H"0-A  '2

%    .0 .%
    %      0 .%
    %    .0 .%        '4"0B  A+P'"       0 .%        1&H"0B  A+P'"
           0  !!
    %    .0 .%        $" 41'   0('  '+"1$2
    %    .0 .%       12+0$-
    %    .0 .%         '+0$-
?? 41'   0('  '+"1$2
```

0 .%0 ¶ à ±ü=óðP10 0 €!À ¥G ß % 0 !! 3 ` ! 0 €!À ` !

0 !!0#) 03 #0%
0 !!0#) 0 .%
0 !!0#) 03 #0% +$ = =A 0 .%

0 !!0#)
0) '4"0-A '2

II & 2 + ! , ! , */ ! . ! ! 5) #
II , 3) "# 3 5) # #Q % 0 % R
II = #) :! #;)) 5) # #% .0 .%

0)
% 0 %
% .0 .%

0 .%
% .0 .%
12+0$ - 0 .%
'+0$ - 0 .%
0 % 0 .%

II & 2 2) ! 0 ±”Mð° % À ÿRß .% À 0 ÿCß 3• .0 ‹ÿ ßI

% 0 ! (' =1+$ $2
% 0 ! 1 ' 2 "141 ' (' =1+$ $2 II ! . /!
% 0 ! 1 ' 2 "141 ') 0! (' =1+$ $2

) 0!
+$ = =A 1 ' 2 "141 '

* *

! 0 % 0 ! 2

% 0 ! ! 0 % 0 !

! 0 % 0 !

0 !

! 0 !

% 0 !

(@€ % ! 0 !

0 !

12"
N12"
p 1B$$
p 1 $+
1 $B-
'+
B '+
B '+
B '+
1 '+ , '+

1(A= - ' +NB'
1(A= - ' A AL
N(A= - '
N(A= - '
N(A= - ' +NB'
N(A= - ' A AL
S <3LER)!<S
$S <3LER)!<S
9S <3LER)!<S

% 0 ! % 0 % 0) ! 0! (' =1+$ $2

0) ! 0!

* *

* *

* *

- Replace *03 #0 %* and *0 0 30 %* #ith the e2istin& rule .

!

S+++(: %onditional Gump parameter @ , , **B** , ;**B** A must be a boolean

S+++*: Operator not supported 'or operand t.pes @e.&. mat* Q vec(A

S+++/:

!

L+++(: -oo man. verte2 input values

L+++*: -oo man. verte2 output values

L+++/: -oo man. uni'orm values

L+++0: -oo man. 'ra&ment output values

L+++8: 4ra&ment shader uses an input #here there is no correspondin& verte2 output

. ! !

. ! !

- : on;s=uare matrices o' t.pe mat%2R consume the same space as a s=uare matri2 o' t.pe mat : #here : is the &reater o' % and R. 1ariables o' t.pe mat) occupies) complete ro#s. -hese rules allo# implementations more 'le2ibilit. in ho# variables are stored.
  Other variables consume onl. the minimum space re=uired.

yÁ y€

Option: : o, this #ould be e2pensive to implement on devices that do not nativel. support it.

RESOL9 -$O : : : o. -he speci'ication should allo# e''icient implementation o' mediump 'loat on 10;bit 'loatin& point hard#are but must also be implementable on devices #hich onl. nativel. support ();bit 'loatin& point.

Should the 'ra&ment shader have a de'ault precisionC

1erte2 shaders have a de'ault hi&h precision because lo#er precisions are not su''icient 'or the ma&orit. o' &raphics applications. , o#ever, man. 'ra&ment shader operations do not bene'it 'rom hi&h precision and

!! !

!! !

2 + - + %

Should local 'unctions hide all 'unctions o' the same same

-his is considered use'ul i' local 'unction declarations are allo#ed. , o#ever, the onl. use 'or local 'unction declarations in GLSL ctic0sips Q1Q l,ctions hide oons eR tG tr • lr G" ltb lQ sEptinsESll 63 ulrations are

!! !

0 6

0

1 >2A +")" %)4 < +")"2

6

!! !

! ! !

RESOL 9 - $O : : Keep the basic preprocessor as de'ined in the GLSL ES 1.++ speci'ication.

4

Option: Replace the current version directive mechanism #ith a b.te or character se=uence that must al#a.s occur at the start o' the shader. -his is similar to other standards that have multiple versions e.&. , - - 3.

Option: <a7e 9 - 4;5 characters an optional 'eature o' GLSL ES 1.++

RESOL9 -$O : : Replace the version directive in GLSL ES 1.+ #ith a character se=uence that must al#a.s occur at the start o' the shader.

5

Option 1: Speci'. the ma2imum as 5 *;vectors. $t is then up to the application to pac7 var.in&s. Other lan&ua&es re=uire the pac7in& to be done b. the application. ! evelopers have not reported this as a problem.

!! !

!! !

RESOL 9 - $O : : Option 1.

%an unde'ined values be made invariantC

$' a t.pe is implemented b. a lar&er native t.pe and due to lac7 o' initiali Eation, a variable o' that t.pe has

$mplementations must usuall. be able to evaluate constant e2pressions at compile time since the. can be

!! !

!! !

E2ample:

0

> hat compiler trans'orms should be allo#edC

!! !

5 9 : /

!! !

! : 1

!! !

```
        IGJ
            1A    3 2            LL
    I J 1 A2A
```

Option): Si&ned inte&er. -his allo#s &reater 'le2ibilit. in calculatin& arra. indices #ithout the need 'or t.pe conversions e.&.

```
        I)#\%J
  222
           1  2         5 =     "     ;     2 )#\%   ;        <

           V1 A                              <
       I     J
```

RESOL9 -$O : : Option). -he principle is that inte&ers that represent values and hence ma. 'orm part o'

-here are some valid uses 'or aliasin&. n ƒuber shaderƒ @i.e. a lar&e shader that consists o' multiple selectable smaller shadersA mi&ht have too man. verte2 inputs i' the. all have uni=ue locations but could

!4

!! !

```
M  I=J 1          I:JI=J
```

Option (:

!! !

! . 1 $

/ 0 & * !