APPENDIX B1

GRID DIAGNOSTIC FUNCTIONS

The following describes the computation of GEMPAK grid diagnostic functions.

Each grid in a grid file is identified by a parameter name, time, level, and vertical coordinate. A scalar grid is a single grid, while a vector grid is composed of two grids containing the u and v components.

The parameter name is used to retrieve a grid from the file, with a few exceptions: Certain special parameters will be computed from other data in the grid file if the parameter name itself is not found in the grid file. These special scalar parameters are

TMPK	DWPK	TVRK	MIXR*	THTA*	DRCT	TMWK*
TMPC	DWPC	TVRC	SMXR	STHA	SPED	TMWC
TMPF	DWPF	TVRF	MIXS	THTE*	RELH	TMWF
		THES*	SMXS	STHE		

where * indicates names which also may be used as operators. Mixing ratio will be computed automatically from dewpoint temperature, specific humidity or vapor pressure, if a pressure grid exists.

The stability indices will be computed automatically from temperature, dewpoint temperature, and wind speed and direction. These special scalar parameters are

CTOT VTOT TOTL KINX SWET

Haines Indices for fire weather detection will be computed automatically from temperature and dewpoint at three different levels. These scalar parameters are:

LHAN	Low elevation Haines Index
MHAN	Middle elevation Haines Index
HHAN	High elevation Haines Index

The Heat Index, HEAT, will also be automatically computed from the temperature and relative humidity.

In addition, precipitation will be converted from inches (I) to millimeters (M) and vice versa, if the grids are named P__M or P__I. The middle numeric characters give the time interval over which the precipitaion accumulated. For example, P24M is a 24-hour precipitation total.

The units for sea surface temperature (SST_), maximum temperature (TMX_) and minimum temperature (TMN_) will be converted automatically. The underscore may be K, C or F.

These special scalar parameter names denote constant value grids:

DTR	**Conversion factor for degrees to radians = PI / 180**	
E	**Base of natural logarithms**	**= 2.71828182**
GRAVTY	**Gravitational constant**	**= 9.80616 (note spelling)**
KAPPA	**Gas constant/specific heat**	**= 2/7**
PI		**= 3.14159265**
RTD	**Conversion factor for radians to degrees = 180 / PI**	
nnn	**Any number (i.e., 2, -10.2, ...)**	

Another class of special parameter names provides information at grid points depending on the navigation of the grid file:

CORL	Coriolis force = 2. * OMEGA * SIN (LATR)
LATR	Latitude in radians
LONR	Longitude in radians
XVAL	Value of the x coordinate in graph coordinates
YVAL	Value of the y coordinate in graph coordinates
MSFX	Map scale factor in the x direction
MSFY	Map scale factor in the y direction
LAND	Land array; land=1, sea=RMISSD
SEA	Sea array; sea=1, land=RMISSD

Finally, scalar grids may be identified by their location within the grid file. The grid number must be prefixed with the symbol #. Note that grids may be renumbered as grids are added to or deleted from the file.

Vector grids are two separate grids containing the u and v components. Special vector parameter names may be used to identify the following vectors:

WND	Total wind
GEO*	Geostrophic wind
AGE*	Ageostrophic wind
ISAL*	Isallobaric wind
THRM*	Thermal wind

where * indicates names that also may be used as operators. Note that all of these wind vectors will have u and v components in meters per second. The total wind must be stored as UWND and VWND in the grid file if the components are north relative and as UREL and VREL if the components are grid relative.

Time, level, and vertical coordinate may be specified in GDATTIM, GLEVEL and GVCORD. However, any of these values may be overridden by in line parameters appended to an operand in the form of ^time@level%ivcord. In-line parameters are only allowed for operands, since they modify parameters for individual grids. The in-line parameters may be entered individually or in combinations in any order.

If more than one file is opened, +n may also be used as an in-line parameter, where n is the number corresponding to the position of the file name entered in GDFILE. If +n is omitted, the first file is used.

Grid operators may be nested, allowing a complicated diagnostic function to be computed. One limitation is that layer and time range operators expect to work on operands read directly from the grid file or computed from special names.

In the following list of diagnostic operators, scalar operands are named Si and vector operands are Vi. Lower case u and v refer to the grid relative components of a vector. All meteorological output grids are in MKS units, except as noted. Operators using PR_ functions are described in the GEMPAK PARAMETER APPENDIX. All scalar and vector differential operators are valid in any map projection for which the map scale factors can be computed. At present, this applies for the stereographic, cylindrical and conic projections available in GEMPAK. In the definitions below, only the cartesian form of the operators is shown. The general curvilinear coordinate forms involving the scale factors are not given.

The operators which are designated for use in polar coordinates are specific to that coordinate system.

SCALAR OUTPUT GRID

Algebraic and trignometric functions (angles are expressed in radians):

ABS **Absolute value**
ABS (S)

ACOS **Arc cosine function**
ACOS (S)

ASIN **Arc sine function**
ASIN (S)

ATAN **Arc tangent function**
ATAN (S)

ATN2 **Arc tangent function**
ATN2 (S1, S2) = ATAN (S1 / S2)

COS **Cosine function**
COS (S)

EXP **Exponential to real**
EXP (S1, S2) = S1 ** S2

EXPI **Exponential to integer**

EXP (S1, S2) = S1 ** NINT (S2)

LN **Natural logarithm**
LN (S) = LOG (S)

LOG **Base 10 logarithm**
LOG (S) = LOG10 (S)

NINT **Round to integer**
NINT (S)

SIN **Sine function**
SIN (S)

SQRT **Square root**
SQRT (S)

TAN **Tangent function**
TAN (S)

ADD **Addition**
ADD (S1, S2) = S1 + S2

MUL **Multiplication**
MUL (S1, S2) = S1 * S2

QUO **Division**
QUO (S1, S2) = S1 / S2

SUB **Subtraction**
SUB (S1, S2) = S1 - S2

SLT **Less than function**
SLT (S1, S2) = S1 < S2

SLE **Less than/equal to**
SLE (S1, S2) = S1 <= S2

SGT **Greater than function**
SGT (S1, S2) = S1 > S2

SGE **Greater than/equal to**
SGE (S1, S2) = S1 >= S2

SBTW **Between function**
SBTW (S1, S2, S3) = S1 > S2 AND S1 < S3

BOOL **Boolean function**
BOOL (S)

MASK **Masking function**
MASK (S1, S2) = RMISSD IF S2 = RMISSD, = S1 otherwise

MISS **Missing value replace**
MISS (S1, S2) = S2 if S1 = RMISSD, = S1 otherwise

ADV **Advection**

ADV (S, V) = - (u * DDX (S) + v * DDY (S))

AVG **Average**
AVG (S1, S2) = (S1 + S2) / 2

AVOR **Absolute vorticity**
AVOR (V) = VOR (V) + CORL

BVSQ **Brunt-Vaisala frequency squared in a layer**
BVSQ (THTA) = [GRAVTY * LDF (THTA)] / [LAV (THTA) * DZ] in PRES coordinates

= -(RDGAS / GRAVTY) * LAV (THTA) *(LAV (PRES) / 1000) ** KAPPA * LDF (PRES) / LAV (PRES) in THTA coordinates

DZ = change in height across the layer

CROS **Vector cross product magnitude**
CROS (V1, V2) = u1 * v2 - u2 * v1

DDEN **Density of dry air (kg / m**3)**
DDEN (PRES, TMPC) = PR_DDEN (PRES, TMPC)

DDR **Partial derivative with respect to R**
DDR (S) is computed using centered finite differences, with backward or forward differences at the boundary. Polar coordinates are assumed, and (R, THETA) maps into (X, Y).

DDT **Time derivative**
DDT (S) = (S (time1) - S (time2)) / (time1 - time2) where the time difference is in seconds.

DDX **Partial derivative with respect to X**
DDX (S) is computed using centered finite differences, with backward or forward differences at the boundary.

DDY **Partial derivative with respect to Y**
DDX (S) is computed using centered finite differences, with backward or forward differences at the boundary.

DEF **Total deformation**
DEF (V) = (STR (V) ** 2 + SHR (V) ** 2) ** .5

DIRN **North relative direction of a vector**
DIRN (V) = PR_DRCT (UN (V), VN (V))

DIRR **Grid relative direction of a vector**
DIRR (V) = PR_DRCT (u, v)

DIV **Divergence**
DIV (V) = DDX (u) + DDY (v)

DOT **Vector dot product**
DOT (V1, V2) = u1 * u2 + v1 * v2

DTH **Partial derivative with respect to THETA**
DTH (S) is computed using centered finite differences, with backward or forward differences at the boundary. Polar coordinates are assumed, and (R, THETA) maps into (X, Y).

FCNT **Coriolis force at the center of a polar coordinate grid**
FCNT (S) can be computed only for lat/lon grids which have been mapped to polar (R,THETA) coordinates and or which the center lat/lon

have been stored with each grid.

FOSB **Fosberg index, also called Fire Weather Index.**
FOSB (TMPC, RELH, SPED) is computed with an empirical formula using surface temperature, relative humidity, and wind speed at the 2 meter or 10 meter level, or the mix of the two. High values indicate high flame lengths and rapid drying.

FRNT **Frontogenesis (K / 100 km / 3 h)**
FRNT (THTA, V) = 1/2 * CONV * MAG (GRAD (THTA)) * (DEF * COS (2 * BETA) - DIV)

CONV = unit conversion factor = 1.08E4 * 1.E5

BETA = ASIN ((- COS (DELTA) * DDX (THTA) -

SIN (DELTA) * DDY (THTA) /

MAG (GRAD (THTA)))

DELTA = 1/2 ATAN (SHR / STR)

GWFS **Horizontal smoothing using normally distributed weights**
GWFS (S,N) with theoretical response of 1/e for N * delta-x wave. Increasing N increases the smoothing.

HIGH **Relative maxima over a grid**
HIGH (S, RADIUS) where RADIUS defines a square region of grid points. The region is a moving search area in which all points are compared to derive a relative maximum.

JCBN **Jacobian determinant**
JCBN (S1, S2) = DDX (S1) * DDY (S2) - DDY (S1) * DDX (S2)

KNTS **Convert meters / second to knots**
KNTS (S) = PR_MSKN (S) = S * 1.9438

LAP **Laplacian operator**
LAP (S) = DIV (GRAD (S))

LAV **Layer average (2 levels)**
LAV (S) = (S (level1) + S (level2)) / 2.

LDF **Layer difference (2 levels)**
LDF (S) = S (level1) - S (level2)

LOWS **Relative minima over a grid**
LOWS (S, RADIUS) where RADIUS defines a square region of grid points. The region is a moving search area in which all points are compared to derive a relative minimum.

MAG **Magnitude of a vector**
MAG (V) = PR_SPED (u, v)

MASS **Mass per unit volume in a layer**
MASS = 100 * LDF (PRES) / (GRAVTY * (level1 - level2))

The 100 converts mb to Pascals. Level1 and level2 are also converted to Pascals when VCOORD = PRES. The volume is expressed in units of m * m * (units of the vertical coordinate). This is an operand.

MDIV **Layer-average mass divergence**

MDIV (V) = DIV ([MASS * LAV (u), MASS * LAV (v)])

MIXR **Mixing ratio**
MIXR (DWPC, PRES) = PR_MIXR (DWPC, PRES)

The units are kg/kg internally, but g/kg on output.

MRAD **Magnitude of storm relative radial wind**
MRAD (V, LAT, LON, DIR, SPD) = DOT (Vrel, R)

where Vrel is the velocity minus the storm motion vector specified by DIR and SPD, and R is a unit vector tangent to a great circle arc from the storm center specified by LAT, LON to a grid point.

MSDV **Layer-average mass-scalar flux divergence**
MSDV (S, V) = DIV ([MASS * LAV (S) * LAV (u), MASS * LAV (S) * LAV (v)])

Note: MASS is computed using the in-line parameter values for V rather than those for S.

MSFC **Psuedo angular momentum (for cross sections)**
MSFC (V) = NORMV (V) + CORL * DIST

DIST is the distance along the cross section in meters. The units for the M-surface are expressed in m/s.

MTNG **Magnitude of storm relative tangential wind**
MTNG (V, LAT, LON, DIR, SPD) = DOT (Vrel, k X R)

where Vrel is the velocity minus the storm motion vector specified by DIR and SPD, and R is a unit vector tangent to a great circle arc from the storm center specified by LAT, LON to a grid point. k denotes the local vertical unit vector.

NORM **Scalar vector component normal to a cross section**
NORM (V) = DOT (V, unit normal vector)

If the starting point for the cross section is on the left, the unit normal vector points into the cross section plane.

PLAT **Latitude at each point in polar coordinates**
PLAT (S)

Note: only the header, which contains the center latitude and longitude, is used.

PLCL **Pressure of the lifting condesation level**
PLCL (PRES, TMPC, DWPC) = PR_PLCL (TMPC, PRES, TLCL)

Note: The temperature of the LCL must be calculated as an intermidiate quantity.

PLON **Longitude at each point in polar coordinates**
PLON (S)

Note: only the header, which contains the center latitude and longitude, is used.

POIS **Solve Poisson eqn. of a forcing function with the given boundary values**
POIS (S1, S2) where S1 is the forcing function grid and S2 is the boundary value.

The equation LAP (POIS) = S1 is solved for POIS.

POLF **Coriolis force at each point in polar coordinates**

POLF (S)

Note: only the header, which contains the center latitude and longitude, is used.

PVOR **Potential vorticity in a layer**
PVOR (S, V) = - GRAVTY * AVOR (VLAV (V)) * LDF (THTA) / (100 * LDF (PRES))

The 100 converts millibars to Pascals.

Units are Kelvins / meters / Pascals / seconds**3 (note that GRAVTY is included). PVOR works on a layer

in PRES or THTA coordinates. In isobaric coordinates, the scalar operand, S, is THTA, THTE, or THES. In isentropic coordinates, the scalar operand, S, is PRES. Multiplying by 10**6 gives standard PV units.

RELH **Relative humidity**
RELH (TEMP, DWPT) = PR_RELH (TEMP, DWPT)

RICH **Richardson stability number in a layer**
RICH (V) = GRAVTY * DZ * LDF (THTA) / (LAV (THTA) * MAG (VLDF (V)) ** 2)

Note: DZ = change in height across the layer.

RICH can be evaluated in PRES, THTA or HGHT vertical coordinate.

ROSS **Rossby number**
ROSS (V1, V2) = MAG (INAD (V1, V2)) / (CORL * MAG (V1))

SAVG **Average over whole grid**
SAVG (S) = average of all non-missing grid point values

SAVS **Average over subset grid**
SAVS (S) = average of all non-missing grid point values in the subset area

SDIV **Flux divergence of a scalar**
SDIV (S, V) = S * DIV (V) + DOT (V, GRAD (S))

SHR **Shear deformation**
SHR (V) = DDX (v) + DDY (u)

SM5S **Smooth scalar grid using a 5-point smoother**
SM5S (S) = .5 * S (i,j) + .125 * (S (i+1,j) + S (i,j+1) + S (i-1,j) + S (i,j-1))

SM9S **Smooth scalar grid using a 9-point smoother**
SM5S (S) = .25 * S (i,j) + .125 * (S (i+1,j) + S (i,j+1) + S (i-1,j) + S (i,j-1)) + .0625 * (S (i+1,j+1) + S (i+1,j-1) + S (i-1,j+1) + S (i-1,j-1))

STAB **Thermodynamic stability within a layer (lapse rate)**
STAB (TMPC) = LDF (TMPC) / DZ in PRES coordinates.

= - (RDGAS / GRAVTY) * LAV (THTA) * (LAV (PRES) / 1000) ** KAPPA *

LDF (PRES) / LAV (PRES) in THTA coordinates

DZ = change in height across the layer.

Units are degrees / kilometer.

STR **Stretching deformation**
STR (V) = DDX (u) - DDY (v)

TANG **Scalar vector component tangential to a cross section**

TANG (V) = DOT (V, unit tangent vector)

If the starting point for the cross section is on the left, the unit tangent vector points to the right.

TAV **Time average (2 times)**
TAV (S) = (S (time1) + S (time2)) / 2.

TDF **Time difference (2 times)**
TDF (S) = S (time1) - S (time2)

THES **Saturated equivalent potential temperature in Kelvin**
TTHES (PRES, TMPC) = PR_THTE (PRES, TMPC, TMPC)

THTA **Potential temperature in Kelvin**
THTA (TMPC, PRES) = PR_THTA (TMPC, PRES)

THTE **Equivalent potential temperature in Kelvin**
THTE (PRES, TMPC, DWPC) = PR_THTE (PRES, TMPC, DWPC)

THWC **Wet bulb potential temperature in Celsius**
THWC (PRES, TMPC, DWPC) = PR_THWC (PRES, TMPC, DWPC)

TLCL **Temperature of the lifitng condensation level**
TLCL (TMPC,DWPC) = PR_TLCL (TMPC, DWPC)

TMST **Parcel temperature in Kelvin along a moist adiabat**
TMST (THTE, PRES) = PR_TMST (THTE, PRES, GUESS)

where THTE is the equivalent potential temperature at the input GLEVEL,

PRES is the pressure level at which the parcel temperature is valid, and GUESS is a guess-field calculated automatically.

TMWK **Wet bulb temperature in Kelvin**
TMWK (PRES, TMPK, RMIX) = PR_TMWK (PRES, TMPK, RMIX)

UN **North relative u component**
UN (V) = zonal wind component

UR **Grid relative u component**
UR (V) = u

VN **North relative v component**
VN (V) = meridional wind component

VOR **Vorticity**
VOR (V) = DDX (v) - DDY (u)

VR **Grid relative v component**
VR (V) = v

WNDX **WINDEX (index for microburst potential)**
WNDX (S1, S2, S3, S4) = 2.5 * SQRT (HGHTF * RATIO * (GAMMA**2 - 30 +

MIXRS - 2 * MIXRF))

TMPCS = surface temperature = S1

HGHTF = AGL Height of Freezing level = S2

MIXRS = surface mixing ratio = S3

MIXRF = freezing level mixing ratio = S4

RATIO = MIXRS / 12 if MIXRS < 12, = 1 otherwise

GAMMA = TMPCS / HGHTF

WSHR **Magnitude of the vertical wind shear in a layer**
WSHR (V) = MAG [VLDF (V)] / DZ in PRES coordinates.

= - (RDGAS / GRAVTY) * LAV (THTA) * (LAV (PRES) / 1000) ** KAPPA *

LDF (PRES) / LAV (PRES) in THTA coordinates.

DZ = change in height across the layer

WSHR can be evaluated in PRES, THTA, or HGHT coordinate.

XAV **Average along a grid row**
XAV (S) = (S (X1) + S (X2) + ... + S (KXD)) / KNT

KXD = number of points in row

KNT = number of non-missing points in row

XAV for a row is stored at every point in that row.

In polar coord., XAV is the average along a radial.

XSUM **Sum along a grid row**
XSUM (S) = (S (X1) + S (X2) + ... + S (KXD))

KXD = number of points in row

XSUM for a row is stored at every point in that row. In polar coord., XSUM is the sum along a radial.

YAV **Average value along a grid column**
YAV (S) = (S (Y1) + S (Y2) + ... + S (KYD)) / KNT

KYD = number of points in column

KNT = number of non-missing points in column

YAV for a column is stored at every point in that column. For polar coordinates, YAV is the average around a circle. If the theta coordinate starts at 0 degrees and ends at 360 degrees, the first radial is not used in computing the average.

YSUM **Sum along a grid column**
SUM (S) = (S (Y1) + S (Y2) + ... + S (KYD))

KYD = number of points in column

YSUM for a column is stored at every point in that column. For polar coordinates, YSUM is the sum around a circle. If the theta coordinate starts at 0 degrees and ends at 360 degrees, the first radial is not used in computing the sum.

VECTOR OUTPUT GRID

AGE **Ageostrophic wind**
AGE (S) = [u (OBS) - u (GEO(S)), v (OBS) - v (GEO(S))]

CIRC **Circulation (for cross sections)**
CIRC (V, S) = [TANG (V), S]

DVDX **Partial x derivative of a vector**
DVDX (V) = [DDX (u), DDX (v)]

DVDY **Partial y derivative of a vector**
DVDY (V) = [DDY (u), DDY (v)]

GEO **Geostrophic wind**
GEO (S) = [- DDY (S) * const / CORL, DDX (S) * const / CORL]

const	S	vert coord
GRAVTY ZMSL	none	
GRAVTY HGHT	PRES	
1	PSYM	THTA
100/RO PRES	HGHT	

RO = PR_DDEN (PRES, TMPC)

GRAD **Gradient of a scalar**
GRAD (S) = [DDX (S), DDY (S)]

GWFV **Horizontal smoothing using normally distributed weights**
GWFV (V,N) with theoretical response of 1/e for N * delta-x wave. Increasing N increases the smoothing.

INAD **Inertial advective wind**
INAD (V1, V2) = [DOT (V1, GRAD (u2)),
DOT (V1, GRAD (v2))]

ISAL **Isallobaric wind**
ISAL (S) = [- DDT (v (GEO(S))) / CORL,
DDT (u (GEO(S))) / CORL]

KCRS **Unit vector k cross a vector**
KCRS (V) = [-v, u]

KNTV **Convert meters / second to knots**
KNTV (V) = [PR_MSKN (u), PR_MSKN (v)]

LTRN **Layer-averaged transport of a scalar**
LTRN (S, V) = [MASS * LAV (S) * LAV (u),
MASS * LAV (S) * LAV (v)]
Note: MASS is computed using the in-line parameter values for V rather than those for S.

NORMV **Vector component normal to a cross section.**
NORMV (V) = NORM (V) * unit normal vector

QVEC **Q-vector at a level (K / m / s)**
QVEC (S, V) = [- (DOT (DVDX (V), GRAD (S))),
- (DOT (DVDY (V), GRAD (S)))] where S can be any thermal paramenter, usually THTA.

QVCL **Q-vector of a layer (mb / m / s)**
QVCL (THTA, V) = (1/(D (THTA) / DP)) *
[(DOT (DVDX (V), GRAD (THTA))),
(DOT (DVDY (V), GRAD (THTA)))]

RAD **Storm relative radial wind**
RAD (V, LAT, LON, DIR, SPD) = SMUL (DOT (Vrel, R), R)
where Vrel is the velocity minus the storm motion specified by DIR and SPD, and R is a unit vector tangent to a great circle arc from the storm center specified by LAT, LON to a grid point.

ROT **Coordinate rotation**
ROT (angle, V) = [u * COS (angle) + v * SIN (angle),
-u * SIN (angle) + v * COS (angle)]

SMUL **Multiply a scalar with each component of a vector**
SMUL (S, V) = [S * u, S * v]

SM5V **Smooth vector grid using a 5-point smoother**
SM5V (V) = .5 * V (i,j) + .125 * (V (i+1,j) + V (i,j+1) +
V (i-1,j) + V (i,j-1))

SQUO **Vector division by a scalar.**
SQUO (S, V) = [u / s, v / s]

TANGV **Vector component tangential to a cross section.**
TANGV (V) = TANG (V) * unit tangent vector

THRM **Thermal wind**
THRM (S) = [u (GEO(S)) (level1) - u (GEO(S)) (level2),
v (GEO(S)) (level1) - v (GEO(S)) (level2)]

TNG **Storm relative tangential wind**

TNG (V, LAT, LON, DIR, SPD) = SMUL (DOT (Vrel, k X R), k X R)

where Vrel is the velocity minus the storm motion vector specified by DIR and SPD, and R is a unit vector tangent to a great circle arc from the storm center specified by LAT, LON to a grid point. k denotes the local vertical unit vector.

VADD **Add the components of two vectors**

VADD (V1, V2) = [u1+u2, v1+v2]

VASV **Vector component of V1 along V2**

VASV (V1, V2) = [DOT (V1,V2) / MAG (V2) ** 2] * V2

VAVG **Average over whole grid**

VAVG (V) = average of all non-missing grid point values

VAVS **Average over subset grid**

VAVS (V) = average of all non-missing grid point values in the subset area

VECN **Create a vector grid from two north relative scalar components**

VECN (S1, S2) = [S1, S2]

VECR **Create a vector grid from two grid relative scalar components**

VECR (S1, S2) = [S1, S2]

VLAV **Layer average for a vector**

VLAV (V) = [(u (level1) + u (level2)) / 2.,
(v (level1) + v (level2)) / 2.]

VLDF **Layer difference for a vector**

VLDF (V) = [u (level1) - u (level1),
v (level1) - v (level2)]

VMUL **Multiply the components of two vectors**

VMUL (V1, V2) = [u1*u2, v1*v2]

VQUO **Divide the components of two vectors**

VQUO (V1, V2) = [u1/u2, v1/v2]

VSUB **Subtract the components of two vectors**

VSUB (V1, V2) = [u1-u2, v1-v2]

VLT **Less than function**

VLT (V, S) = V IF |V| < S

VLE **Less than or equal to function**

VLE (V, S) = V IF |V| <= S

VGT **Greater than function**

VGT (V, S) = V IF |V| > S

VGE **Greater than or equal to function**

VGE (V, S) = V IF |V| >= S

VBTW **Between function**

VBTW (V, S1, S2) = V IF S1 < |V| < S2

VMSK **Masking function**

VMSK (V, S) = RMISSD IF S = RMISSD
= V otherwise