tf.contrib.nn.deprecated_flipped_sigmoid_cross_entropy_with_logits

```
deprecated_flipped_sigmoid_cross_entropy_with_logits(
    logits,
    targets,
    name=None
)
```

Defined in `tensorflow/contrib/nn/python/ops/cross_entropy.py`.

Computes sigmoid cross entropy given `logits`.

This function diffs from tf.nn.sigmoid_cross_entropy_with_logits only in the argument order.

Measures the probability error in discrete classification tasks in which each class is independent and not mutually exclusive. For instance, one could perform multilabel classification where a picture can contain both an elephant and a dog at the same time.

For brevity, let `x = logits`, `z = targets`. The logistic loss is

```
  z * -log(sigmoid(x)) + (1 - z) * -log(1 - sigmoid(x))
= z * -log(1 / (1 + exp(-x))) + (1 - z) * -log(exp(-x) / (1 + exp(-x)))
= z * log(1 + exp(-x)) + (1 - z) * (-log(exp(-x)) + log(1 + exp(-x)))
= z * log(1 + exp(-x)) + (1 - z) * (x + log(1 + exp(-x))
= (1 - z) * x + log(1 + exp(-x))
= x - x * z + log(1 + exp(-x))
```

For x < 0, to avoid overflow in exp(-x), we reformulate the above

```
  x - x * z + log(1 + exp(-x))
= log(exp(x)) - x * z + log(1 + exp(-x))
= - x * z + log(1 + exp(x))
```

Hence, to ensure stability and avoid overflow, the implementation uses this equivalent formulation

```
max(x, 0) - x * z + log(1 + exp(-abs(x)))
```

`logits` and `targets` must have the same type and shape.

Args:

- `logits`: A `Tensor` of type `float32` or `float64`.
- `targets`: A `Tensor` of the same type and shape as `logits`.
- `name`: A name for the operation (optional).

Returns:

A `Tensor` of the same shape as `logits` with the componentwise logistic losses.

Raises:

- `ValueError`: If `logits` and `targets` do not have the same shape.

Last updated November 2, 2017.