

BlockGateRS.java

```

package net.minecraft.src;

//Created by MoareAI
//RS/SR Latch (Set/Reset Memory)

import java.util.Random;

public class BlockGateRS extends Block
{
    //Class parameters
    protected BlockGateRS(int i, boolean OutputF, boolean OutputG)
    {
        super(i, 255, Material.circuits);
        field_OutputF = OutputF;
        field_OutputG = OutputG;
        setBlockBounds(0.0F, 0.0F, 0.0F, 1.0F, 0.125F, 1.0F);
    }

    //Use render types
    public boolean renderAsNormalBlock()
    {
        return false;
    }

    //Block type (preset)
    public int getRenderType()
    {
        return 14;
    }

    public boolean canPlaceBlockAt(World world, int i, int j, int k)
    {
        if(!world.isBlockOpaqueCube(i, j - 1, k))
        {
            return false;
        } else
        {
            return super.canPlaceBlockAt(world, i, j, k);
        }
    }

    public boolean canBlockStay(World world, int i, int j, int k)
    {
        if(!world.isBlockOpaqueCube(i, j - 1, k))
        {
            return false;
        } else
        {
            return super.canBlockStay(world, i, j, k);
        }
    }

    //Updates on/off
    public void updateTick(World world, int i, int j, int k, Random random)
    {
        int l = world.getBlockMetadata(i, j, k);
        boolean InputS = func_InputS(world, i, j, k, l);
        boolean InputR = func_InputR(world, i, j, k, l);
        if((InputS || field_OutputF) && !InputR)
        {
            world.setBlockAndMetadataWithNotify(i, j, k,
mod_LogicalGates.GateRSON.blockID, l);
        } else
    }

```

BlockGateRS.java

```

        if(((InputS || field_OutputF) && !InputR) || InputS)
        {
            world.setBlockAndMetadataWithNotify(i, j, k,
mod_LogicalGates.GateSROn.blockID, 1);
        } else
        {
            world.setBlockAndMetadataWithNotify(i, j, k,
mod_LogicalGates.GateRSOff.blockID, 1);
        }
    }

    //Update block
    public void onNeighborBlockChange(World world, int i, int j, int k, int l)
    {
        if(!canBlockStay(world, i, j, k))
        {
            dropBlockAsItem(world, i, j, k, world.getBlockMetadata(i, j, k));
            world.setBlockWithNotify(i, j, k, 0);
            return;
        }
        world.scheduleBlockUpdate(i, j, k, blockID, 0);
    }

    //Signal out
    public boolean isPoweringTo(IBlockAccess iblockaccess, int i, int j, int k, int l)
    {
        if(!field_OutputF)
        {
            return false;
        }
        int i1 = iblockaccess.getBlockMetadata(i, j, k) & 3;
        if((((i1 == 1) && field_OutputF) || ((i1 == 0) && field_OutputG)) && l == 3)
        {
            return true;
        }
        if((((i1 == 2) && field_OutputF) || ((i1 == 1) && field_OutputG)) && l == 4)
        {
            return true;
        }
        if((((i1 == 3) && field_OutputF) || ((i1 == 2) && field_OutputG)) && l == 2)
        {
            return true;
        }
        return (((i1 == 0) && field_OutputF) || ((i1 == 3) && field_OutputG)) && l ==
5);
    }

    //Signal in Back
    //Back
    private boolean func_InputS(World world, int i, int j, int k, int l)
    {
        int i1 = l & 3;
        switch(i1)
        {
            case 0: // '\0'
                return world.isBlockIndirectlyProvidingPowerTo(i, j, k + 1, 3);

            case 2: // '\002'
                return world.isBlockIndirectlyProvidingPowerTo(i, j, k - 1, 2);

            case 3: // '\003'
                return world.isBlockIndirectlyProvidingPowerTo(i + 1, j, k, 5);
        }
    }

```

BlockGateRS.java

```

        case 1: // '\001'
            return world.isBlockIndirectlyProvidingPowerTo(i - 1, j, k, 4);
        }
        return false;
    }

//Right
private boolean func_InputR(World world, int i, int j, int k, int l)
{
    int i1 = l & 3;
    switch(i1)
    {
        case 0: // '\0'
            return world.isBlockIndirectlyProvidingPowerTo(i + 1, j, k, 5);

        case 2: // '\002'
            return world.isBlockIndirectlyProvidingPowerTo(i - 1, j, k, 4);

        case 3: // '\003'
            return world.isBlockIndirectlyProvidingPowerTo(i, j, k - 1, 2);

        case 1: // '\001'
            return world.isBlockIndirectlyProvidingPowerTo(i, j, k + 1, 3);
        }
        return false;
    }

    public void onBlockPlacedBy(World world, int i, int j, int k, EntityLiving
entityliving)
    {
        int l = ((MathHelper.floor_double((double)((entityliving.rotationYaw * 4F) /
360F) + 0.5D) & 3) + 2) % 4;
        world.setBlockMetadataWithNotify(i, j, k, l);
        boolean InputS = func_InputS(world, i, j, k, l);
        if(InputS)
        {
            world.scheduleBlockUpdate(i, j, k, blockID, 1);
        }
    }

    public void onBlockAdded(World world, int i, int j, int k)
    {
        world.notifyBlocksOfNeighborChange(i + 1, j, k, blockID);
        world.notifyBlocksOfNeighborChange(i - 1, j, k, blockID);
        world.notifyBlocksOfNeighborChange(i, j, k + 1, blockID);
        world.notifyBlocksOfNeighborChange(i, j, k - 1, blockID);
        world.notifyBlocksOfNeighborChange(i, j - 1, k, blockID);
        world.notifyBlocksOfNeighborChange(i, j + 1, k, blockID);
    }

    public boolean isOpaqueCube()
    {
        return false;
    }

//Dropped
public int idDropped(int i, Random random)
{
    return mod_LogicalGates.GateRS.shiftedIndex;
}

//Torch particles
public void randomDisplayTick(World world, int i, int j, int k, Random random)

```

